HTML5+CSS3+jQuery Mobile +Bootstrap开发APP 从入门到精通 视频教学版



张工厂 著

掌握Web页面开发技术，快速实现移动APP

- 将最实用的技巧融入每个案例中，教你快速成为开发APP的高手
- 计算器、求职招聘、购物网站、手机游戏4个APP综合案例，快速提升实战能力
- 提供QQ群技术支持，答疑解惑，提高学习效率



清华大学出版社

HTML5+CSS3+jQuery Mobile +Bootstrap开发APP 从入门到精通



视频教学版

张工厂 著

清華大学出版社
北 京

内 容 简 介

本书以应用实例和综合实战案例的形式逐一详解了 HTML5 网页设计的文档结构、文本、图像、用 HTML5 创建超链接、表格、使用表单、HTML5 中的音频和视频、HTML5 绘制图形、地理定位、离线 Web 应用、用 CSS3 设置字体与段落、表格和表单的样式、美化图片、背景、边框和用 CSS3+DIV 布局页面、jQuery Mobile UI 组件、jQuery Mobile 事件、使用最新 Bootstrap 4 框架、开发计算器、开发求职招聘、开发购物网站和开发手机游戏等内容。

通过对本书实例和综合案例的学习与演练，读者可以尽快掌握所学的知识，提高网页设计的实战能力，同时本书在网上提供了实例源代码，可供读者直接查看和调用，以便快速上手或进行二次开发。

本书内容丰富，理论结合实践，适合没有任何移动网站和开发基础的初学者，同时也可作为有一定 HTML5 和 CSS3 基础，想精通移动网站和开发的人员参考书。

图书在版编目（CIP）数据

HTML5+CSS3+jQuery Mobile+Bootstrap 开发 APP 从入门到精通：视频教学版 / 张工厂著. — 北京：清华大学出版社，2019
（Web 前端技术丛书）
ISBN 978-7-302-52723-7

Ⅰ. ①H… Ⅱ. ①张… Ⅲ. ①超文本标记语言－程序设计②网页制作工具③JAVA 语言－程序设计 Ⅳ.①TP312.8②TP393.092.2

中国版本图书馆 CIP 数据核字（2019）第 063182 号

责任编辑：夏毓彦
封面设计：王　翔
责任校对：闫秀华
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张：**34.75　　**字　　数：**890 千字
版　　次：2019 年 6 月第 1 版　　**印　　次：**2019 年 6 月第 1 次印刷
定　　价：99.00 元

产品编号：080575-01

前　言

由于原生应用程序的开发费用比较高，时间也比较长，因此 jQuery Mobile 函数库的出现就很好地解决了这个问题，通过 HTML5 新技术和 jQuery Mobile 搭配使用，开发出的网站与普通没有区别，受到广大客户的欢迎。

本书内容

第 1 章是 HTML5 快速入门。包括 HTML5 的基本概念、HTML5 网页文档结构、HTML5 文件的编写方法和 HTML5 语法的新变化。

第 2 章介绍 HTML5 网页中的文本和图像。包括添加文本、文本排版、文字列表和网页中的图像等。

第 3 章介绍用 HTML5 建立超链接。包括 URL 的概念、超链接标记、创建热点区域和浮动框架 iframe 等。

第 4 章介绍用 HTML5 创建表格。包括表格基本结构及操作、完整的表格标记和制作计算机标价表等。

第 5 章介绍使用表单。包括表单概述、表单基本元素和表单高级元素的使用等。

第 6 章介绍在 HTML5 中的音频和视频。包括<audio>和<video>标记等。

第 7 章介绍 HTML5 绘制图形。包括 canvas 概述、绘制基本形状、绘制渐变图形、绘制变形图形、图形组合、绘制带阴影的图形、使用图像、绘制文字及图形的保存和恢复等。

第 8 章介绍地理定位和离线 Web 应用。包括地理定位、离线 Web 应用和 Web 存储技术。

第 9 章是 CSS3 快速入门。包括 CSS3 基本概念、编写与浏览 CSS3，以及在 HTML5 中使用 CSS3 的方法。

第 10 章介绍 CSS3 中文字与段落属性。主要包括字体属性、文本高级样式和段落属性，最后通过两个综合实例进一步讲述了如何设置公司标题和旅游网页混排的方法和技巧。

第 11 章介绍用 CSS3 设置表格和表单的样式。包括表格基本样式与 CSS3 和表单，并通过 5 个综合案例，进一步讲述表单和表格样式的制作方法和技巧。

第 12 章介绍 CSS3 美化图片。主要包括图片样式、图片的对齐和图文混排等。

第 13 章介绍 CSS3 美化背景与边框。主要包括背景相关属性、边框、圆角边框、图像边框和边距，最后通过一个综合实例进一步讲述了如何制作网上书店圆角边框效果。

第 14 章介绍网页的定位与布局。主要包括定位方式、float 浮动定位、overflow 溢出定位、

visibility 隐藏定位、块和行内元素 display、定位布局新闻页面。

第 15 章介绍 CSS3 盒子和 DIV 布局。主要包括认识 div 层、盒子模型、CSS3 新增弹性盒模型、图文排版效果和淘宝导购菜单。

第 16 章介绍 CSS3+DIV 页面基本排版。主要包括 CSS3 排版概念、固定宽度布局、新增 CSS3 多列布局和电子企业首页。

第 17 章介绍 jQuery Mobile。主要包括认识 jQuery Mobile、跨平台移动设备网页 jQuery Mobile、创建多页面的 jQuery Mobile 网页、将页面作为对话框使用、绚丽多彩的页面切换效果。

第 18 章介绍 jQuery Mobile UI 组件。主要包括套用 UI 组件、列表、面板和可折叠块、导航条、jQuery Mobile 主题。

第 19 章介绍 jQuery Mobile 事件。主要包括页面事件、触摸事件、滚屏事件和定位事件。

第 20 章介绍 Bootstrap 4 框架。主要包括 Bootstrap 概述、下载 Bootstrap、安装 Bootstrap、使用常用组件、胶囊导航选项卡。

第 21 章介绍开发计算机的项目实训。

第 22 章介绍开发求职招聘的项目实训。

第 23 章介绍开发购物网站的项目实训。

第 24 章介绍开发游戏的项目实训。

本书特色

知识全面：知识由浅入深，涵盖了所有 HTML5、CSS3 和 jQuery Mobile 知识点，便于读者循序渐进地掌握 HTML5 +CSS3+jQuery Mobile 网页设计技术。

图文并茂：注重操作，图文并茂，在介绍案例的过程中，每一个操作均有对应的插图。这种图文结合的方式使读者在学习过程中能够直观、清晰地看到操作的过程及效果，便于更快地理解和掌握。

易学易用：颠覆传统“看”书的观念，变成一本能“操作”的图书。

案例丰富：把知识点融汇于系统的案例实训中，并且结合经典案例进行讲解和拓展，进而达到“知其然，并知其所以然”的效果。

贴心周到：对读者在学习过程中可能会遇到的疑难问题以“提示”和“技巧”的形式进行了说明，以免读者在学习过程中走弯路。

技术支持：提供了实例和综合案例的源代码，可让读者在实战应用中掌握网页布局的每一项技能，使本书真正体现“自学无忧”，成为一本物超所值的好书。

读者可加入 QQ 群：941230971 向作者索要源代码、教学幻灯片和精品教学视频。

读者对象

本书是一本完整介绍 HTML5+CSS3+jQuery Mobile 开发的教程，内容丰富、条理清晰、实用性强，适合如下读者学习使用。

- 没有任何移动网站和开发基础的初学者。
- 有一定 HTML5 和 CSS3 基础，想精通移动网站和开发的人员。
- 大专院校及培训学校的老师和学生。

代码、课件与教学视频下载

本书代码、课件与教学视频下载地址为：https://share.weiyun.com/5vLUp8Q。如果下载有问题，请联系 booksaga@163.com，邮件主题为“H5C3MB 代码”。

鸣谢

本书由张工厂著，参加编写工作的还有李小威、刘增产、王秀荣、王天护、王英英、刘增杰、刘玉萍、胡同夫、皮素芹、王猛、王攀登、王婷婷、王朵朵、王维维、张芳、刘玉红等，在此表示感谢。虽然倾注了编者的努力，但由于水平有限、时间仓促，书中难免有错漏之处，欢迎批评指正。

封面照片说明

月亮湾是婺源县城到李坑景区半路的一处免费景点，其恰如月亮的形态、隔岸徽派民居的典雅、周边秀美的景色相互融合，构成了这个如梦如幻的美景。感谢天涯摄影家行摄之无疆友情支持这幅美照。

编　者

2019 年 1 月

目　录

第 1 章　HTML5 快速入门

目前网络已经成为人们生活中不可缺少的一部分，网页设计也成为学习计算机的重要内容之一。制作网页可采用可视化编辑软件，但是无论采用哪一种网页编辑软件，最后都是将所设计的网页转化为 HTML。因为 HTML 是网页的基础语言，所以本章就先来介绍 HTML 的基本概念、编写方法及浏览 HTML 文件的方法，使读者初步了解 HTML，从而为后面的学习打下基础。

1.1　HTML5 概述

由于因特网上的信息是以网页的形式展示给用户的，因此网页是网络信息传递的载体。网页文件是用一种标记语言书写的，这种语言称为 HTML（Hyper Text Markup Language，超文本标记语言）。

HTML 是一种标记语言，而不是一种编程语言，主要用于描述超文本中内容的显示方式。标记语言从诞生到今天，经历了十几载，发展过程中也有很多曲折，经历的版本及发布日期如表 1-1 所示。

表 1-1　HTML 经历的版本

版本	发布日期	说明
超文本标记语言（第一版）	1993 年 6 月	作为互联网工程工作小组（IETF）工作草案发布（并非标准）
HTML 2.0	1995 年 11 月	作为 RFC 1866 发布，在 RFC 2854 于 2000 年 6 月发布之后被宣布已经过时
HTML 3.2	1996 年 1 月 14 日	W3C（万维网联盟）推荐标准
HTML 4.0	1997 年 12 月 18 日	W3C 推荐标准
HTML 4.01	1999 年 12 月 24 日	微小改进，W3C 推荐标准

（续表）

版本	发布日期	说明
ISO HTML	2000 年 5 月 15 日	基于严格的 HTML 4.01 语法，是国际标准化组织和国际电工委员会的标准
XHTML 1.0	2000 年 1 月 26 日	W3C 推荐标准，后来经过修订于 2002 年 8 月 1 日重新发布
XHTML 1.1	2001 年 5 月 31 日	较 1.0 有微小改进
XHTML 2.0 草案	没有发布	2009 年，W3C 停止了 XHTML 2.0 工作组的工作
HTML5	2014 年 10 月	W3C 推荐标准

HTML 是一种标记语言，标记语言经过浏览器的解释和编译，虽然它本身不能显示在浏览器中，但在浏览器中可以正确显示 HTML 标记的内容。HTML 语言从 1.0~5.0 经历了巨大的变化，从单一的文本显示功能到图文并茂的多媒体显示功能，许多特性经过多年的完善，已经成为一种非常成熟的标记语言。尤其是 HTML5，对多媒体的支持功能更强，其新增功能如下：

- 新增语义化标记，使文档结构明确；
- 新的文档对象模型（DOM）；
- 实现 2D 绘图的 Canvas 对象；
- 可控媒体播放；
- 离线存储；
- 文档编辑；
- 拖放；
- 跨文档消息；
- 浏览器历史管理；
- MIME 类型和协议注册。

对于这些新功能，支持 HTML5 的浏览器在处理 HTML 代码错误的时候会更灵活，而那些不支持 HTML5 的浏览器将忽略 HTML5 代码。

HTML5 不是一种编程语言，而是一种描述性的标记语言，用于描述超文本中的内容和结构。HTML 最基本的语法是<标记符></标记符>。标记符通常都是成对使用，有一个开头标记和一个结束标记。结束标记只是在开头标记的前面加一个斜杠“/”。当浏览器收到 HTML 文件后，就会解释里面的标记符，然后把标记符相对应的功能表达出来。

如在 HTML 中用<p></p>标记符来定义一个段落，用
标记符来定义一个换行符。当浏览器遇到<p></p>标记符时，会把该标记中的内容自动形成一个段落。当遇到
标记符时，

会自动换行，并且该标记符后的内容会从一个新行开始。这里的
标记符是单标记，没有结束标记，标记后的“/”符号可以省略，为了规范代码，一般建议加上。

1.2 HTML5 的文档结构

HTML5 文档最基本的结构主要包括文档类型说明、文档开始标记、元信息、主体标记和页面注释标记。

1.2.1 文档类型说明

HTML5 设计准则中最重要的一条是化繁为简，Web 页面的文档类型说明（Doctype）被极大地简化了。

在 HTML4 或早期的版本中，创建 HTML 文档时，文档头部的类型说明代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

上面为 XHTML 文档类型说明，读者可以看到这段代码既麻烦又难记，HTML5 对文档类型进行了简化，简单到 15 个字符就可以了，代码如下：

```
<!DOCTYPE html>
```



Doctype 申明需要出现在 HTML 文件的第一行。

1.2.2 HTML 标记

HTML 标记代表文档的开始，由于 HTML 语言语法的松散特性，该标记可以省略，但是为了使之符合 Web 标准和文档的完整性，用户要养成良好的编写习惯，建议不要省略该标记。

HTML 标记以<html>开头，以</html>结尾，文档的所有内容书写在开头和结尾的中间部分。语法格式如下：

```
<html>
...
</html>
```

1.2.3 头标记 head

头标记 head 用于说明文档头部相关信息，一般包括标题信息、元信息、定义 CSS 样式和脚本代码等。HTML 的头部信息是以<head>开始，以</head>结束，语法格式如下：

```
<head>
…
</head>
```



<head>元素的作用范围是整篇文档，定义在 HTML 文档头部的内容往往不会在网页上直接显示。

1. 标题标记 title

HTML 页面的标题一般是用来说明页面的用途的，它显示在浏览器的标题栏中。在 HTML 文档中，标题信息设置在<head>与</head>之间。标题标记以<title>开始，以</title>结束，语法格式如下：

```
<title>
…
</title>
```

在标记中间的“… ”就是标题的内容，它可以帮助用户更好地识别页面。预览网页时，设置的标题在浏览器的左上方标题栏中显示（如图 1-1 所示）。页面的标题只有一个，位于 HTML 文档的头部，即<head>和</head>之间。

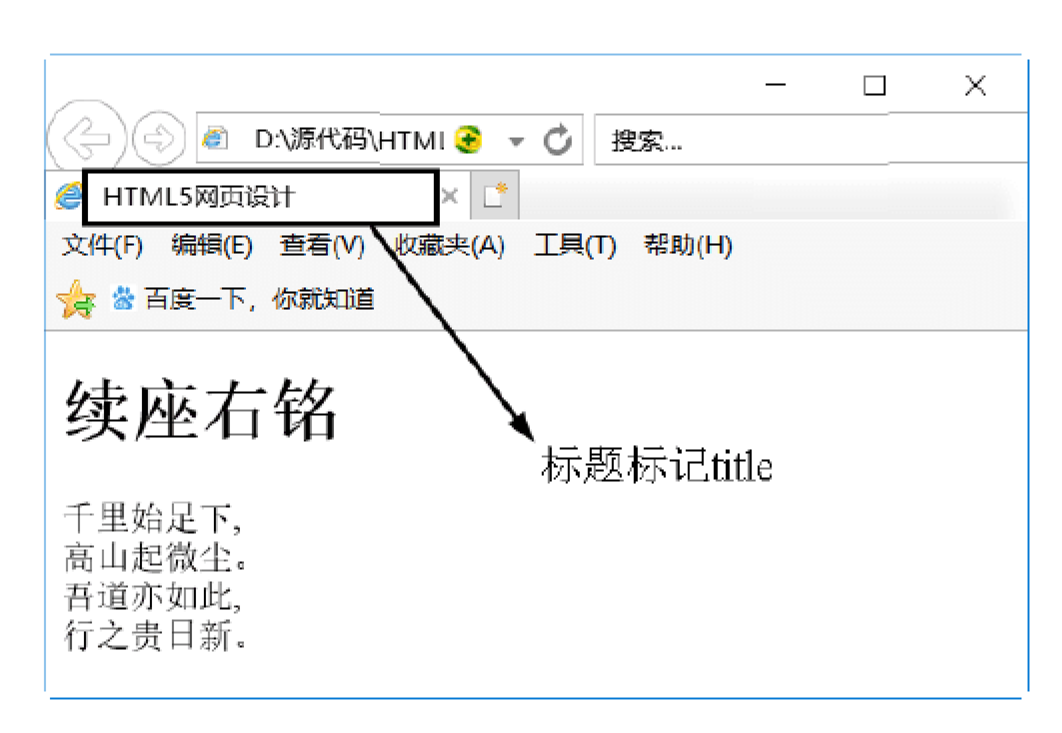


图 1-1　标题栏在浏览器中的显示效果

2. 元信息标记 meta

<meta>标记可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。

<meta>标记位于文档的头部，不包含任何内容。<meta>标记的属性定义了与文档相关联的名称/值，<meta>标记提供的属性及取值如表 1-2 所示。

表 1-2 <meta>标记提供的属性及取值

属性	值	描述
charset	character encoding	定义文档的字符编码
content	some_text	定义与 http-equiv 或 name 属性相关的元信息
http-equiv	content-type expires refresh set-cookie	把 content 属性关联到 HTTP 头部
name	author description keywords generator revised others	把 content 属性关联到一个名称

（1）字符集 charset 属性

在 HTML5 中有一个新的 charset 属性，它使字符集的定义更加容易。例如，下列代码告诉浏览器，网页使用“ISO-8859-1”字符集显示。

```
<meta charset="ISO-8859-1">
```

（2）搜索引擎的关键字

在早期，meta keywords 关键字对搜索引擎的排名算法起到一定的作用，也是很多人进行网页优化的基础。关键字在浏览时是看不到的，使用格式如下：

```
<meta name="keywords" content="关键字,keywords" />
```

说明：

- 不同的关键字之间应用半角逗号隔开（英文输入状态下），不要使用“空格”或“|”间隔；
- 是 keywords，不是 keyword；
- 关键字标签中的内容应该是一个个的短语，而不是一段话。

例如，定义针对搜索引擎的关键词，代码如下：

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript" />
```

关键字标记 Keywords，曾经是搜索引擎排名中很重要的因素，但现在已经被很多搜索引擎完全忽略。如果我们加上这个标记对网页的综合表现没有坏处，不过，如果使用不恰当的话，对网页非但没有好处，还有欺诈的嫌疑。在使用关键字标记 Keywords 时，要注意以下几点：

- 关键字标记中的内容要与网页核心内容相关，确信使用的关键词出现在网页文本中。
- 使用用户易于通过搜索引擎检索的关键字，过于生僻的词汇不太适合做 meta 标记中的关键字。
- 不要重复使用关键字，否则可能会被搜索引擎惩罚。
- 一个网页的关键字标记里最多包含 3~5 个最重要的关键字，不要超过 5 个。
- 每个网页的关键字应该不一样。



由于设计者或 SEO 优化者以前对 Meta Keywords 关键字的滥用，导致目前它在搜索引擎排名中的作用很小。

（3）页面描述

meta description（描述元标记）是一种 HTML 元标记，用来简略描述网页的主要内容，通常被搜索引擎用于搜索结果页上展示给最终用户看的一段文字片段。页面描述在网页中是不显示出来，页面描述的使用格式如下：

```
<meta name="description" content="网页的介绍" />
```

例如，定义对页面的描述，代码如下：

```
<meta name="description" content="免费的 Web 技术教程。" />
```

（4）页面定时跳转

使用<meta>标记可以使网页在经过一定时间后自动刷新，可以通过将 http-equiv 属性值设置为 refresh 来实现。content 属性值可以设置为更新时间。

在浏览网页时经常会看到一些欢迎信息的页面，在经过一段时间后，这些页面会自动转到其他页面，这就是网页的跳转。页面定时刷新跳转的语法格式如下：

```
<meta http-equiv="refresh" content="秒;[url=网址]" />
```

说明：上面的“[url=网址]”部分是可选项，如果有这部分，则页面定时刷新并跳转；如果省略该部分，则页面只定时刷新，不进行跳转。

例如，实现每 5 秒刷新一次页面，将下述代码放入 head 标记部分即可。

```
<meta http-equiv="refresh" content="5" />
```

1.2.4 网页的主体标记 body

网页所要显示的内容都放在网页的主体标记内，它是 HTML 文件的重点所在，后面章节所要介绍的 HTML 标记都将放在这个标记内。然而它并不仅仅是一个形式上的标记，它本身也可以控制网页的背景颜色或背景图像，这会在后面进行介绍。主体标记是以<body> 开始，以</body>结束，语法格式如下：

```
<body>
…
</body>
```

注意，在构建 HTML 结构时，标记不允许交错出现，否则会造成错误。

例如，在下列代码中，<body>开始标记出现在<head>标记内。

```
<html>
<head>
<title>标记测试</title>
<body>
</head>
</body>
</html>
```

代码中的第 4 行<body>开始标记和第 5 行的</head>结束标记出现了交叉，这是错误的。HTML 中的所有标记都是不允许交错出现的。

1.2.5 页面注释标记<!-- -->

注释是在 HTML 代码中插入的描述性文本，用来解释该代码或提示其他信息。注释只出现在代码中，浏览器对注释代码不进行解释，并且在浏览器的页面中不显示。在 HTML 源代码中适当地插入注释语句是一种非常好的习惯。对于设计者日后的代码修改、维护工作很有好处。另外，如果将代码交给其他设计者，其他人也能很快读懂前者所撰写的内容。

语法格式如下：

```
<!--注释的内容-->
```

注释语句元素有前后两半部分，前半部分由一个左尖括号、一个半角感叹号和两个连字符组成，后半部分由两个连字符和一个右尖括号组成。

```
<html>
<head>
<title>标记测试</title>
</head>
<body>
<!-- 这里是标题-->
<h1>HTML5从入门到精通</h1>
</body>
</html>
```

页面注释不但可以对 HTML 中一行或多行代码进行解释说明，而且还能注释掉这些代码。如果希望某些 HTML 代码在浏览器中不显示，则可以将这部分内容放在<!--和-->之间，例如修改上述代码：

```
<html>
<head>
<title>标记测试</title>
</head>
<body>
<!--
<h1>HTML5从入门到精通</h1>
-->
</body>
</html>
```

修改后的代码，将<h1>标记作为注释内容处理，在浏览器中将不会显示这部分内容。

1.3　HTML5 文件的编写方法

有两种方式来产生 HTML 文件：一种是自己写 HTML 文件，事实上这并不是很困难，也不需要特别的技巧；另一种是使用 HTML 编辑器，它可以辅助使用者来做编写的工作。

1.3.1　使用记事本手工编写 HTML 文件

前面介绍到 HTML5 是一种标记语言，标记语言代码是以文本形式存在的，因此所有的记事本工具都可以作为它的开发环境。HTML 文件的扩展名为.html 或.htm，将 HTML 源代码输入到记事本并保存之后，可以在浏览器中打开文档以查看其效果。

使用记事本编写 HTML 文件，具体操作步骤如下：

01 单击 Windows 桌面上的【开始】按钮，选择【所有程序】➢【附件】➢【记事本】命令，打开一个记事本，在记事本中输入 HTML 代码，如图 1-2 所示。

02 编辑完 HTML 文件后，选择【文件】➢【保存】命令或按 Ctrl+S 组合键，在弹出的

【另存为】对话框中选择【保存类型】为【所有文件】，并将文件扩展名设为.html 或.htm，如图 1-3 所示。

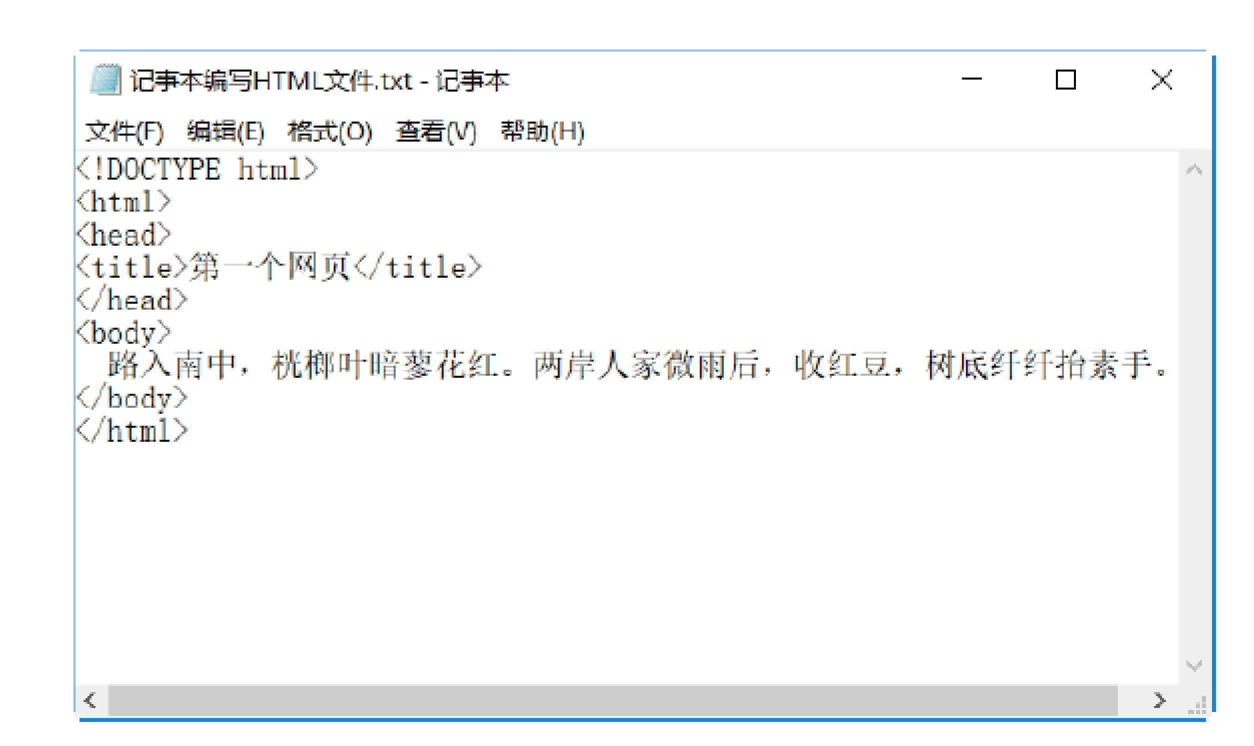


图 1-2 编辑 HTML 代码

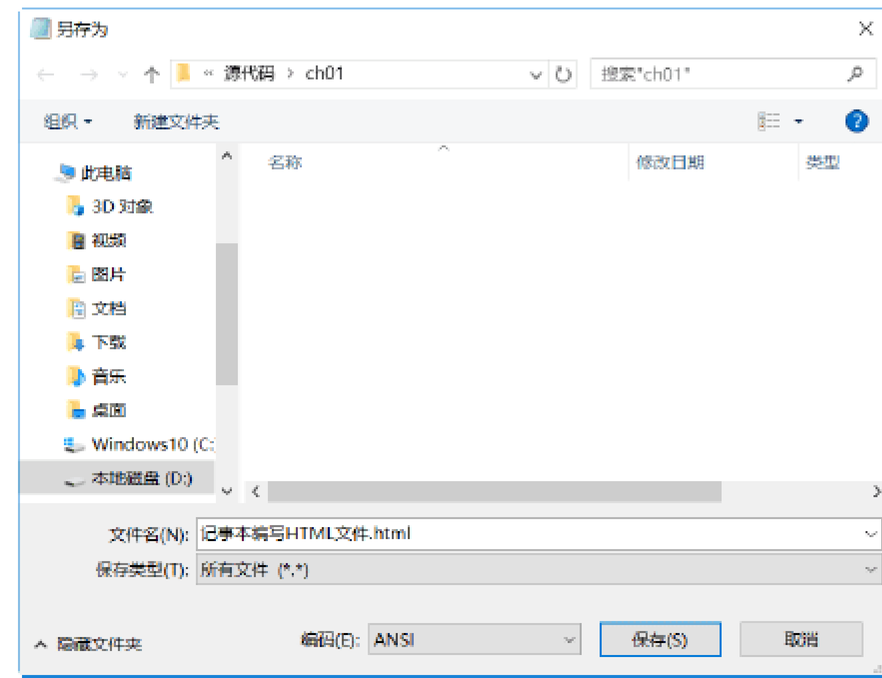


图 1-3 【另存为】对话框

03 单击【保存】按钮，保存文件。打开网页文档，在 IE 浏览器中预览效果，如图 1-4 所示。

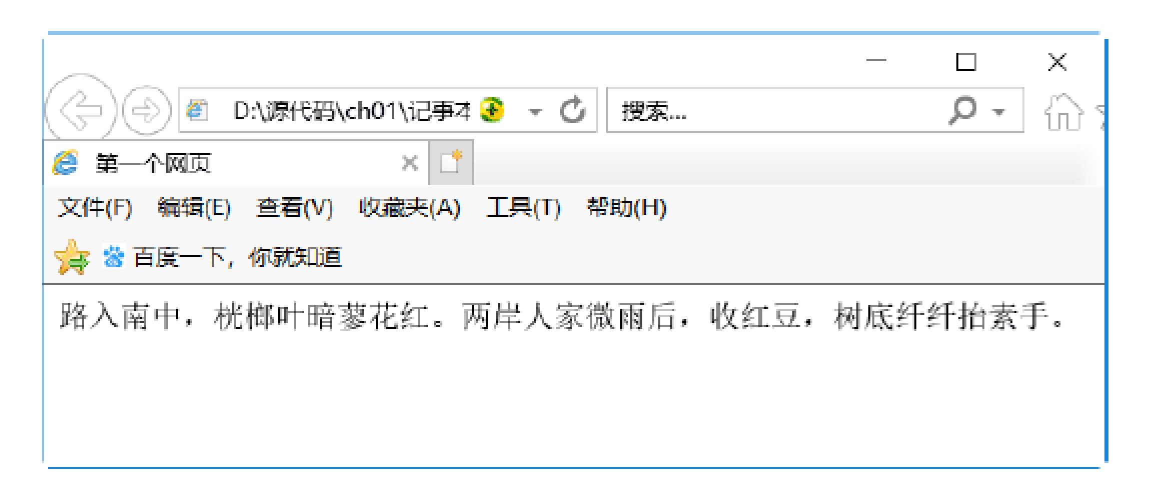


图 1-4 网页的浏览效果

1.3.2 使用 Dreamweaver CC 编写 HTML 文件

常言道："工欲善其事，必先利其器"。虽然使用记事本可以编写 HTML 文件，但是编写效率太低，对于语法错误及格式都没有提示，因此可以使用专门编写 HTML 网页的工具来弥补这种缺陷。Adobe 公司的 Dreamweaver CC 用户界面十分友好，是一款非常优秀的网页开发工具，深受广大用户的喜爱。

使用 Dreamweaver CC 编写 HTML 文件，具体操作步骤如下：

01 启动 Dreamweaver CC（如图 1-5 所示），在欢迎屏幕的【新建】栏中选择 HTML 选项，或者执行菜单中的【文件】➢【新建】命令（快捷键 Ctrl+N）。

02 弹出【新建文档】对话框（如图 1-6 所示），在【页面类型】选项区中选择 HTML 选项，【文档类型】选择【HTML5】选项。



图 1-5　包含欢迎屏幕的主界面



图 1-6　“新建文档”对话框

03 单击【创建】按钮，创建 HTML 文件，如图 1-7 所示。

04 在文档工具栏中单击【代码】按钮，切换到代码视图，如图 1-8 所示。

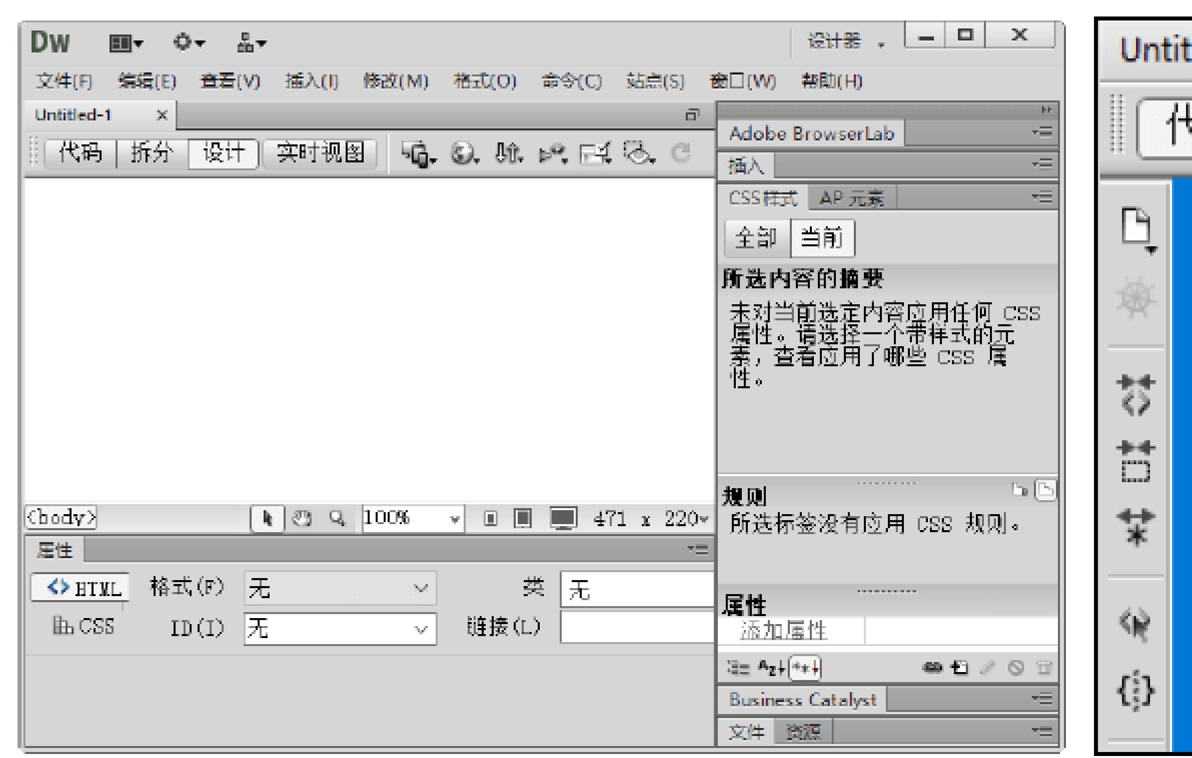


图 1-7　设计视图下显示创建的文件

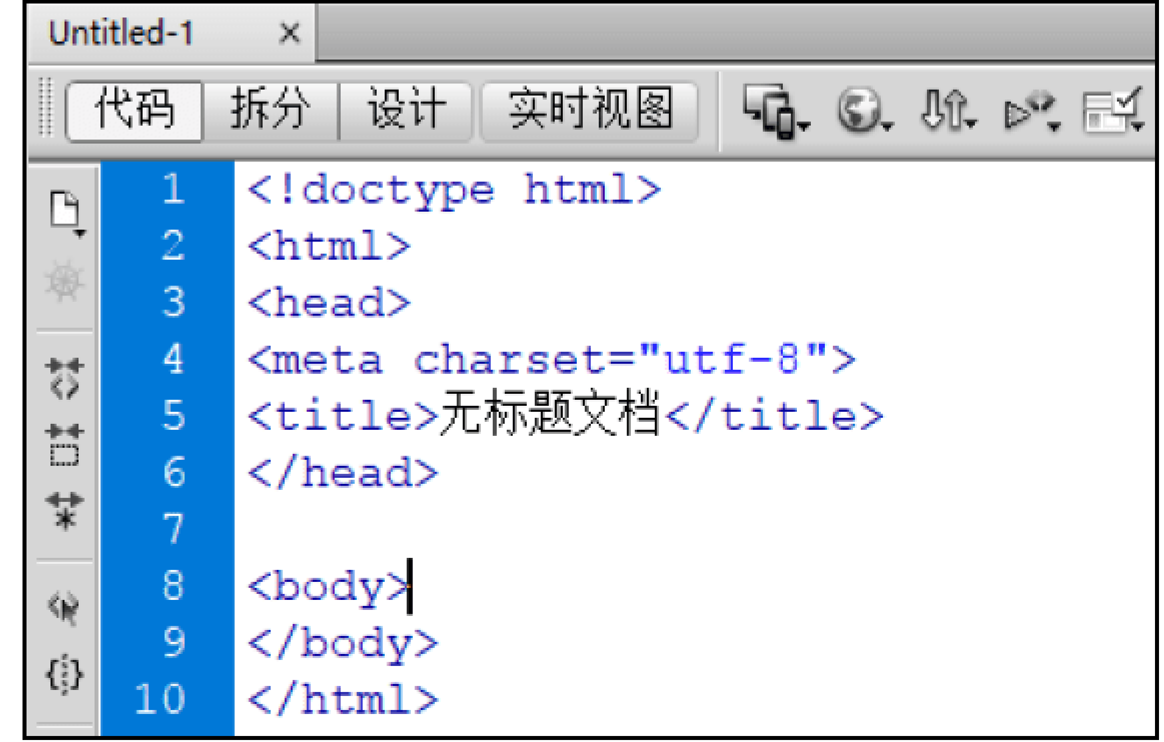


图 1-8　代码视图下显示创建的文件

05 修改 HTML 文件标题，将代码<title>标记中的“无标题文档”修改成“蝶恋花”。

06 在<body>标记中键入“溪上平岗千叠翠，万树亭亭，争作拏云势。”，完整的 HTML 代码如下所示。

```
<!DOCTYPE html>
<head>
<meta charset=utf-8" />
<title>第一个网页</title>
</head>
<body>
溪上平岗千叠翠，万树亭亭，争作拏云势。
</body>
</html>
```

07 保存文件。执行菜单中的【文件】➢【保存】命令或按 Ctrl+S 组合键，弹出【另存为】对话框，选择保存位置并输入文件名，单击【保存】按钮，如图 1-9 所示。

08 单击文档工具栏的图标，或者按下功能键 F12 使用默认浏览器查看网页，预览效果如图 1-10 所示。

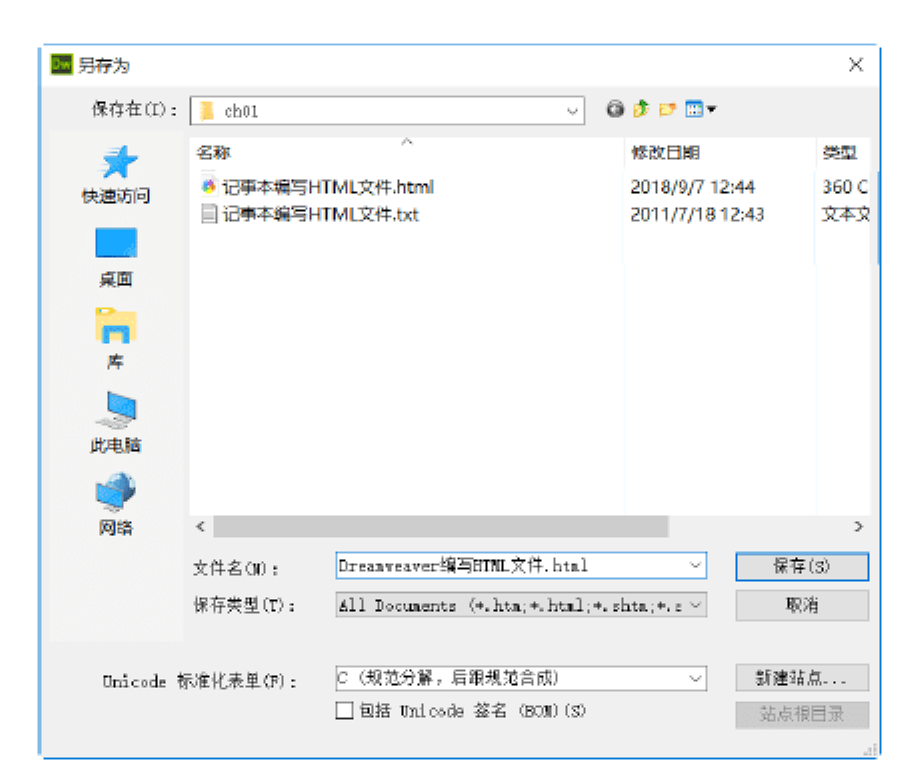


图 1-9　保存文件

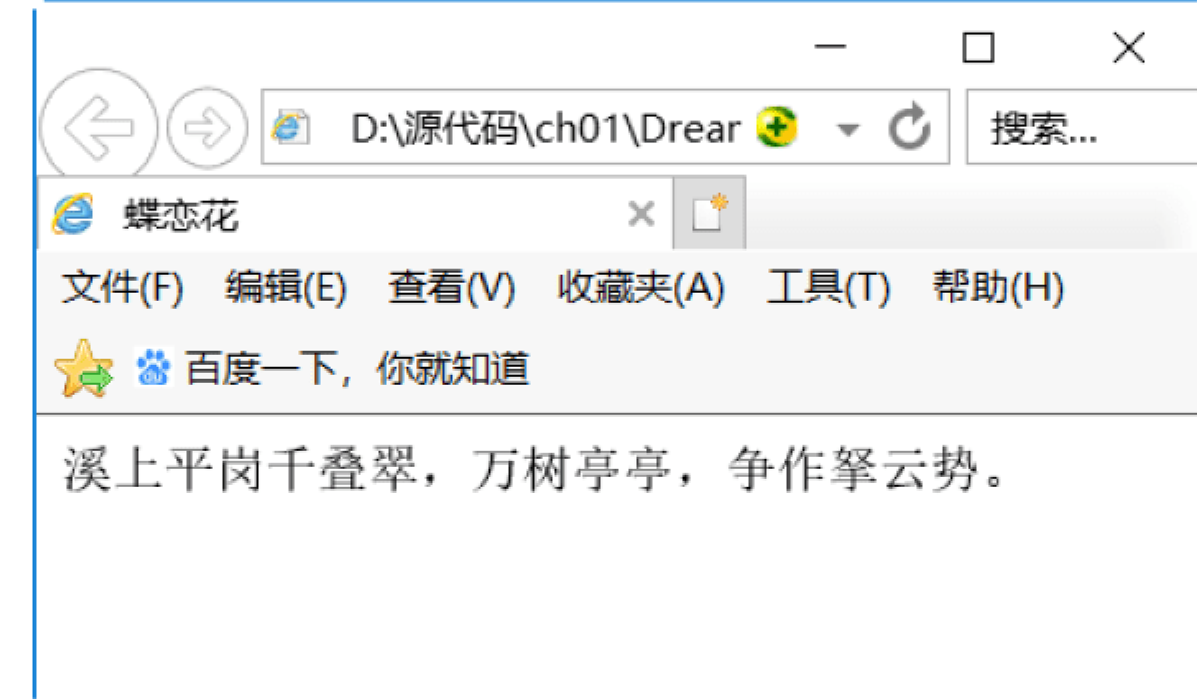


图 1-10　浏览器预览效果

1.4　HTML5 语法的新变化

为了兼容各个不统一的页面代码，HTML5 的设计在语法方面做了以下变化。

1.4.1　标签不再区分大小写

标签不再区分大小写是 HTML5 语法变化的重要体现。例如：

```
<!DOCTYPE html>
<html>
<head>
<title>不再区别大小写标签</title>
</head>
<BODY>
人到情多情转薄，而今真个不多情。
</body>
</html>
```

在 IE 11.0 浏览器中预览，效果如图 1-11 所示。

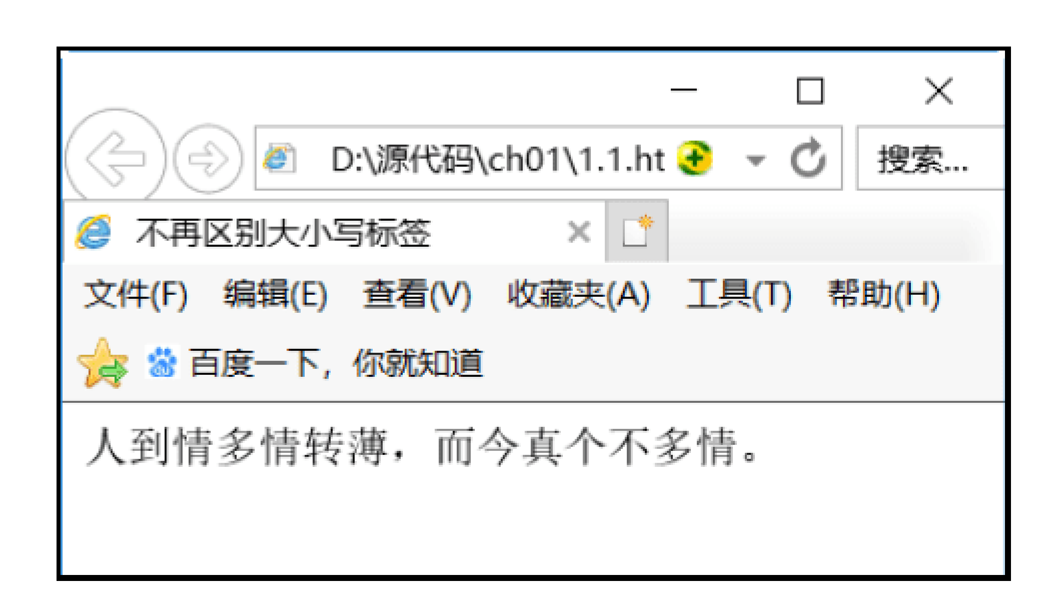


图 1-11　网页预览效果

虽然“<BODY>人到情多情转薄，而今真个不多情。</body>”中开始标记和结束标记不匹配，但是这完全符合 HTML5 的规范。用户可以通过 W3C 提供的在线验证页面来测试上面的网页，验证网址为 http://validator.w3.org/。

1.4.2 允许属性值不使用引号

在 HTML5 中，属性值不放在引号中也是正确的。例如以下代码片段：

```
<input checked="a" type="checkbox"/>
<input readonly type="text"/>
<input disabled="a" type="text"/>
```

上述代码片段与下面的代码片段效果是一样的：

```
<input checked=a type=checkbox/>
<input readonly type=text/>
<input disabled=a type=text/>
```

注意，尽管 HTML5 允许属性值可以不使用引号，但是仍然建议读者加上引号。因为如果某个属性的属性值中包含空格等，则很容易引起混淆的属性值，此时可能会引起浏览器的误解。例如以下代码：

```
<img src=ss ch01/1.1.jpg />
```

此时浏览器就会误以为 src 属性的值就是 ss，这样将无法解析路径中的 01.jpg 图片。如果想正确解析到图片的位置，就只有添加上引号。

1.4.3 允许部分属性值的属性省略

在 HTML5 中，部分标志性属性的属性值可以省略。例如以下代码是完全符合 HTML5 规则的：

```
<input checked type="checkbox"/>
<input readonly type="text"/>
```

其中 checked="checked"省略为 checked，而 readonly="readonly"省略为 readonly。

在 HTML5 中，可以省略属性值的属性如表 1-3 所示。

表 1-3　可以省略属性值的属性

属　性	省略属性值
checked	省略属性值后，等价于 checked="checked"
readonly	省略属性值后，等价于 readonly="readonly"
defer	省略属性值后，等价于 defer="defer"
ismap	省略属性值后，等价于 ismap="ismap"
nohref	省略属性值后，等价于 nohref="nohref"
noshade	省略属性值后，等价于 noshade="noshade"
nowrap	省略属性值后，等价于 nowrap="nowrap"
selected	省略属性值后，等价于 selected="selected"
disabled	省略属性值后，等价于 disabled="disabled"
multiple	省略属性值后，等价于 multiple="multiple"
noresize	省略属性值后，等价于 noresize="noresize"

1.5　专家解惑

问题 1：如何理解 HTML5 中的单标记和双标记书写方法？

HTML5 中的标记分为单标记和双标记。所谓单标记是指没有结束标记的标记，双标记是指既有开始标记又包含结束标记。

对于不允许写结束标记的单标记元素，只可以使用“<元素 />”的形式进行书写。例如“
…</br>”的书写方式是错误的，正确的书写方式为
。当然，在 HTML5 之前的版本中
这种书写方法可以被沿用。HTML5 中不允许写结束标记的元素有 area、base、br 、col、command、embed、hr、img、input、keygen、link、meta、param、source、track、wbr。

对于部分双标记可以省略结束标记。HTML5 中允许省略结束标记的元素有 li、dt、dd、p、rt、rp、optgroup、option、colgroup、thead、tbody、tfoot、tr、td、th。

HTML5 中有些元素还可以完全被省略标记，即使这些标记被省略了，该元素还是以隐式的方式存在的。HTML5 中允许省略全部标记的元素有 html、head、body、colgroup、tbody。

问题 2：为何使用记事本编辑 HTML 文件无法在浏览器中预览，而是直接在记事本中打开？

很多初学者在保存文件时，没有将 HTML 文件的扩展名.html 或.htm 作为文件的后缀，导

致文件还是以.txt 为扩展名，因此无法在浏览器中查看。如果读者是通过单击右键创建记事本文件的，那么在给文件重命名时一定要以 html 或.htm 作为文件的后缀。特别要注意的是，当 Windows 系统的扩展名是隐藏时，更容易出现这样的错误。读者可以在【文件夹选项】对话框中设置是否显示扩展名。

问题 3：在网页中，语言的编码方式有哪些？

在 HTML5 网页中，<meta>标记的 charset 属性用于设置网页的内码语系，也就是字符集的类型，国内常用的是 GB 码，通常设置为“GB2312”（简体中文）和“UTF-8”两种。英文是“ISO-8859-1”字符集，此外还有其他的字符集，这里不再介绍。

第2章　HTML5网页中的文本和图像

文本和图像是网页中非常重要且经常使用的元素，本章将讲解如何在网页中使用文本、文本结构标记及图像的方法。

2.1　添加文本

网页中的文本可以分为两大类：一类是普通文本；另一类是特殊字符文本。

2.1.1　普通文本

所谓普通文本是指汉字或在键盘上可以输出的字符。读者可以在 Dreamweaver CC 代码视图的<body>标记部分直接输入，或者在设计视图下直接输入。

如果有现成的文本，则可以使用复制的方法把其他窗口中需要的文本复制过来。在粘贴文本的时候，如果只希望把文字粘贴过来，而不需要粘贴其文档中的格式，则可以使用 Dreamweaver CC 的“选择性粘贴”功能。

“选择性粘贴”功能只在 Dreamweaver CC 的设计视图中起作用，因为在代码视图中，粘贴的仅是文本，不会有格式。例如，将 Word 文档表格中的文字复制到网页中，而不需要表格结构，其操作方法为：执行【编辑】➢【选择性粘贴】命令或按快捷键 Shift+Ctrl+V，弹出【选择性粘贴】对话框，选中【仅文本】单选按钮，如图 2-1 所示。

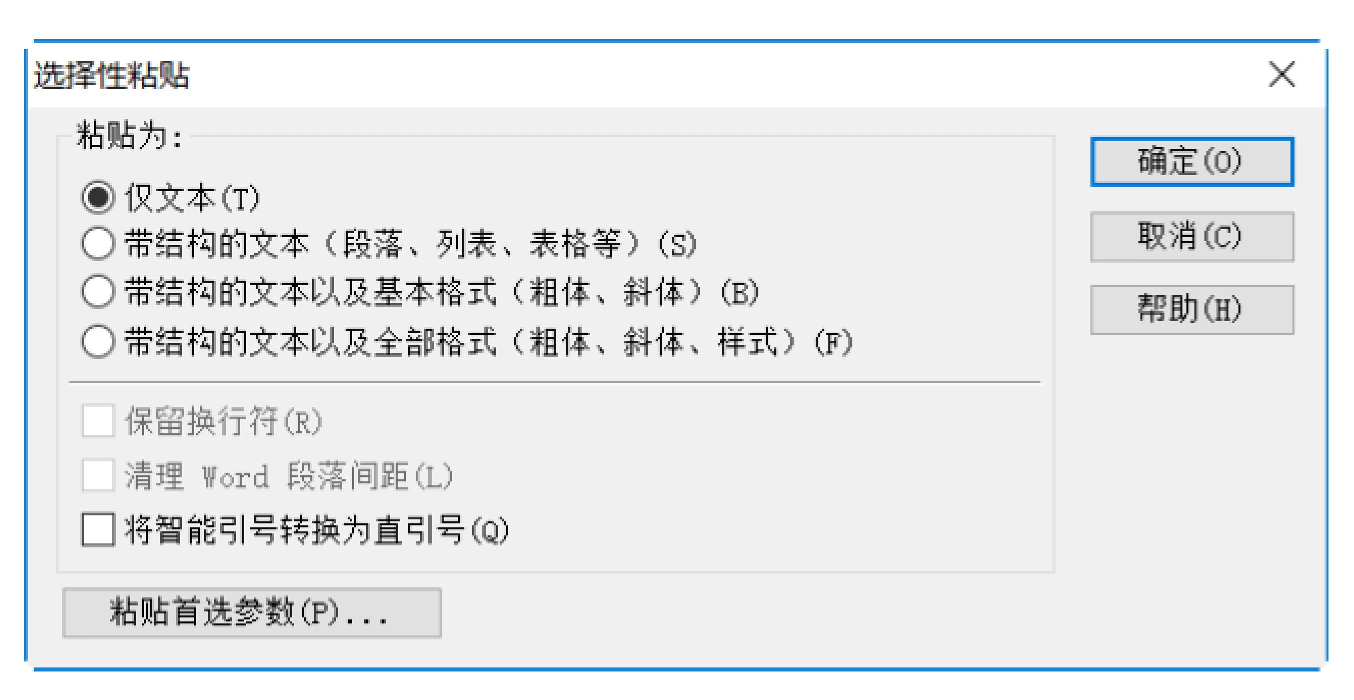


图 2-1　【选择性粘贴】对话框

2.1.2 特殊文字符号

目前，很多行业上的信息都出现在网络上，每个行业都有自己的行业特性，如数学、物理和化学都有特殊的符号。如何在网页上显示这些特殊字符是本小节将要讲述的内容。

在 HTML 中，特殊字符以“&”开头，以“；”结尾，中间为相关字符编码。例如，大括号和小括号被用于声明标记，因此如果在 HTML 代码中出现“<”和“>”符号，就不能直接输入了，需要当作特殊字符处理。在 HTML 中，用“&lt；”代表符号“<”，用“&gt；”代表符号“>”。如输入公式 a>b，在 HTML 中需要这样表示：a>b。

HTML 中还有大量这样的字符，如空格、版权等。常用特殊字符如表 2-1 所示。

表 2-1 常用特殊字符

显示	说明	HTML 编码
	半角大的空白	
	全角大的空白	
	不断行的空白格	
<	小于	<
>	大于	>
&	&符号	&
"	双引号	"
©	版权	©
®	已注册商标	®
™	商标（美国）	™
×	乘号	×
÷	除号	÷

在编辑化学公式或物理公式时，使用特殊字符的频度非常高。如果每次输入时都去查询或者要记忆这些特殊特号的编码，那么工作量是相当大的，在此作者为读者提供了以下技巧：

（1）在 Dreamweaver CC 的设计视图下输入字符，如输入 a>b 这样的表达式。对于部分键盘上没有的字符可以借助“中文输入法”的软键盘。在中文输入法的软键盘上单击鼠标右键，弹出特殊类别项（如图 2-2 所示），选择所需类型，如选择“数学符号”，弹出数学相关符号（如图 2-3 所示），单击“÷”按钮即可输入。

P C 键盘　标点符号
希腊字母　数字序号
俄文字母　数学符号
注音符号　单位符号
拼　音　制表符
日文平假名　特殊符号
日文片假名
极品五笔

图 2-2　特殊符号分类



图 2-3　数学符号

（2）文字与文字之间的空格，如果超过一个，就从第 2 个空格开始，都会被忽略掉。快捷地输入空格的方法如下：将输入法切换成“中文输入法”，并置于“全角”（Shift+空格）状态，直接按键盘上的空格键即可。

（3）对于上述两种方法都无法实现的字符，可以使用 Dreamweaver CC 的【插入】菜单实现。选择【插入】➢HTML➢【特殊字符】菜单命令，在所需要的字符中选择，如果没有所需要的字符，就选择【其他字符】选项。



尽量不要使用多个“ ”来表示多个空格，因为多数浏览器对空格的距离实现是不一样的。

2.1.3　文本特殊样式

在文档中经常会出现重要文本（加粗显示）、倾斜文本、上标和下标文本等。

1. 重要文本

重要文本通常以粗体显示、强调方式显示或加强调方式显示，HTML 中的<b>标记、<em>标记和<strong>标记分别用来实现这三种显示方式。

【例 2.1】（实例文件：ch02\2.1.html）

```
<!DOCTYPE html>
<html>
<head>
<title>无标题文档</title>
</head>
<body>
<!--<b>标记、<em>标记和<strong>标记-->
<p><b>我是粗体文字</b> </p>
<p><em>我是强调文字</em> </p>
<p><strong>我是加强调文字</strong></p>
</body>
</html>
```

在 IE 11.0 中预览效果如图 2-4 所示，实现了文本的三种显示方式。

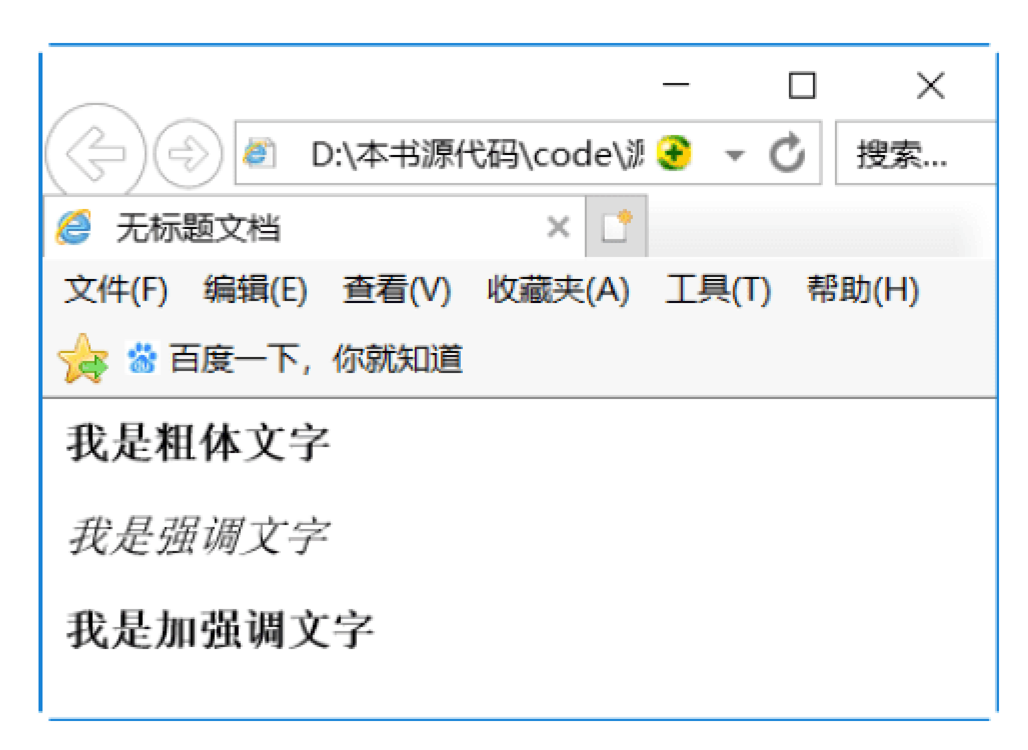


图 2-4 重要文本预览效果

2. 倾斜文本

HTML 中的<i>标记实现了文本的倾斜显示。放在<i></i>之间的文本将以斜体显示。

【例 2.2】（实例文件：ch02\2.2.html）

```
<!DOCTYPE html>
<html>
<head>
<title>无标题文档</title>
</head>
<body>
<i>我将会以斜体字显示</i>
</body>
</html>
```

在 IE 11.0 中预览效果如图 2-5 所示，其中文字以斜体显示。

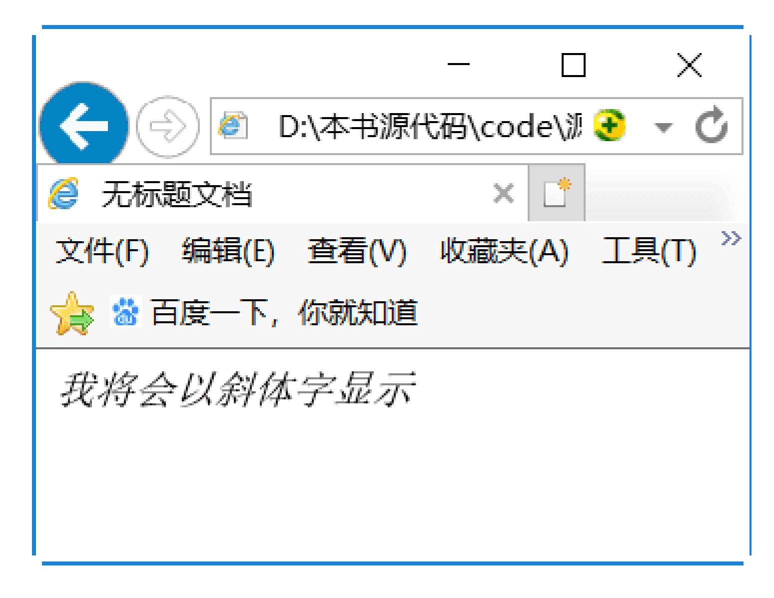


图 2-5 倾斜文本预览效果



HTML 中的重要文本和倾斜文本标记已经过时，是需要读者忘记的标记，这些标记都应该使用 CSS 样式来实现。随着后面学习的深入，读者会逐渐发现，即使 HTML 和 CSS 实现相同的效果，但是 CSS 所能实现的控制远远比 HTML 要细致、精确得多。

3. 上标和下标文本

在 HTML 中用<sup>标记实现上标文字，用<sub>标记实现下标文字。<sup>和<sub>都是双标记，放在开始标记和结束标记之间的文本会分别以上标或下标形式出现。

【例 2.3】（实例文件：ch02\2.3.html）

```
<!DOCTYPE html>
<html>
<head>
<title>无标题文档</title>
</head>
<body>
 <!--上标显示-->
 <p>c=a<sup>2</sup>+b<sup>2</sup></p>
<!--下标显示-->
 <p>H<sub>2</sub>+O→H<sub>2</sub>O</p>
</body>
</html>
```

在 IE 11.0 中预览效果如图 2-6 所示，分别实现了上标和下标文本显示。

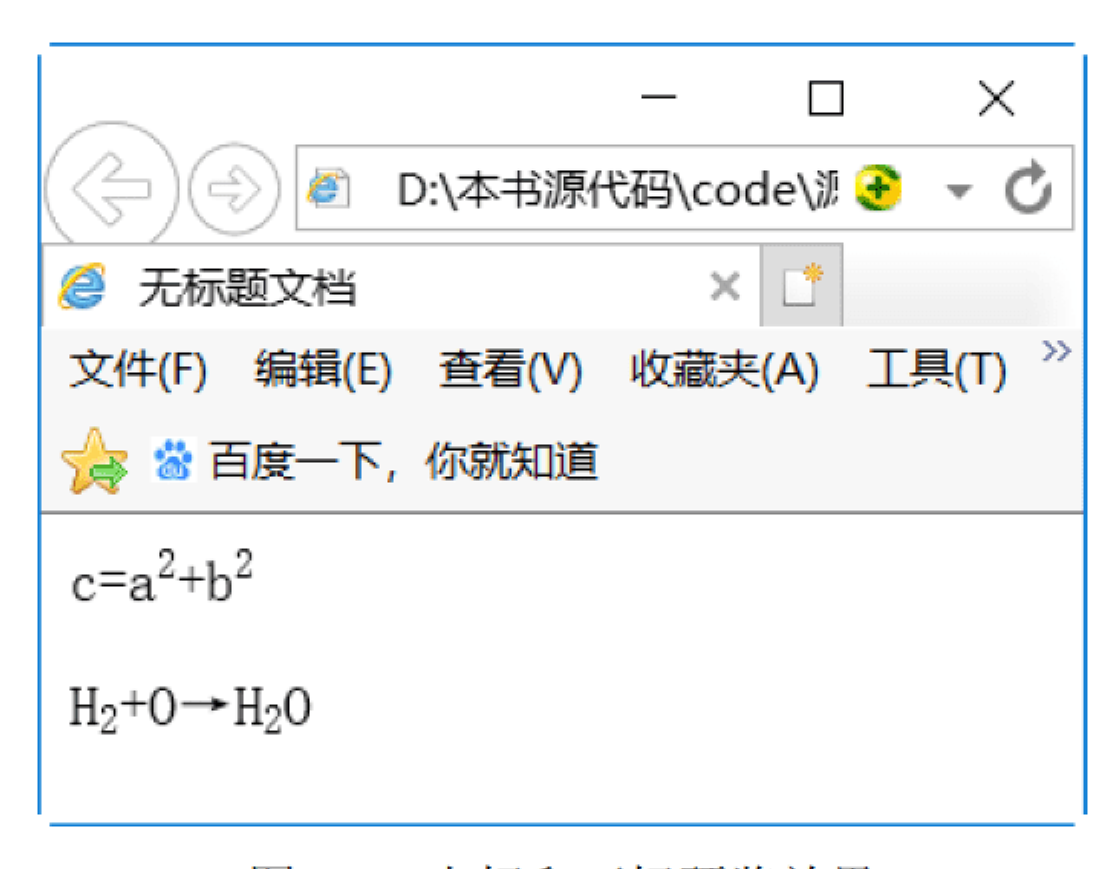


图 2-6　上标和下标预览效果

2.2　文本排版

在网页中，如果要把文字都合理地显示出来，就离不开段落标记的使用。对网页中文字段落进行排版，并不像文本编辑软件 Word 那样可以定义许多模式来安排文字的位置。在网页中要让某一段文字放在特定的地方是通过 HTML 标记来完成的。

2.2.1　换行标记
与段落标记<p>

浏览器在显示网页时，完全按照 HTML 标记来解释 HTML 代码，忽略多余的空格和换行。

在 HTML 文件中，不管输入多少空格（按空格键）都将被视为一个空格，换行（按 Enter 键）也是无效的。在 HTML 中，换行使用
标记，换段使用<p>标记。

1. 换行标记

换行标记
是一个单标记，它没有结束标记，是英文单词 break 的缩写，作用是将文字在一个段内强制换行。一个
标记代表一个换行，连续的多个标记可以实现多次换行。使用换行标记时，在需要换行的位置添加
标记即可。例如，下面的代码实现了对文本的强制换行。

【例 2.4】（实例文件：ch02\2.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>文本段换行</title>
</head>
<body>
本节目标<br/>网页中的文字是如何设置的<br/>如何在Dreamweaver中处理文字<br/>如何对文本进行格式化（CSS）<br/>熟悉使用Dreamweaver进行样式表的创建与应用
</body>
</html>
```

虽然在 HTML 源代码中，主体部分的内容在排版上没有换行，但是增加
标记后，在 IE 11.0 中预览效果如图 2-7 所示，实现了换行效果。

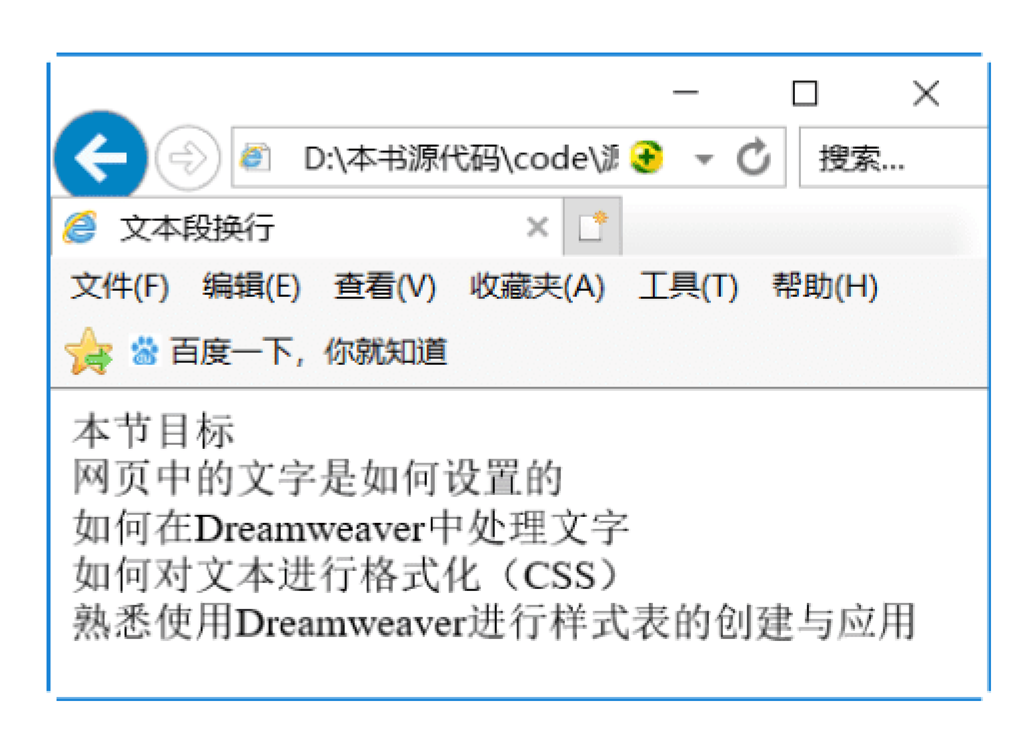


图 2-7 换行标记的使用

2. 段落标记<p>

段落标记是双标记，即<p></p>，在<p>开始标记和</p>结束标记之间的内容形成一个段落。如果省略结束标记，就从<p>标记开始，直到遇见下一个段落标记之前的文本，都在一段段落内。段落标记中的 p 是英文单词 paragraph（段落）的首字母，用来定义网页中的一段文本，文本在一个段落中会自动换行。

【例 2.5】（实例文件：ch02\2.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>段落标记的使用</title>
</head>
<body>
 <p>HTML5、CSS3应用教程之 与DIV说Bey!Bey!</p>
<p>Web设计师可以使用HTML4和CSS2.1完成一些很酷的东西。我们可以在不使用陈旧的基于&lt;table &gt;布局的基础上完成文档逻辑结构并创建内容丰富的网站。我们可以在不使用内联&lt; font &gt;和&lt; br &gt;标记的基础上对网站添加漂亮而细腻的风格样式。事实上，我们目前的设计能力已经让我们远离了那个可怕的浏览器战争时代、专有协议和那些充满闪动、滚动和闪烁的丑陋网页。
<p>
<p>
虽然我们现在已经普遍使用了HTML4和CSS2.1，但是我们还可以做得更好！我们可以重组代码的结构并能让页面代码更富有语义化特性。我们可以缩减带给页面美丽外观样式的代码量并让它们有更高的可扩展性。现在，HTML5和CSS3正跃跃欲试地等待大家，下面就让我们来看看它们是否真的能让我们的设计提升一个高度吧。
</p>
<p>
曾经，设计师们经常会频繁使用基于&lt; table &gt;的没有任何语义的布局。不过最终还是要感谢像Jeffrey Zeldman和Eric Meyer这样的思想革新者，聪明的设计师们慢慢地接受了相对更语义化的&lt; div &gt;布局替代了&lt; table &gt;布局，并且开始调用外部样式表。但不幸的是，复杂的网页设计需要大量不同的标记结构代码，我们把它叫作“&lt; div &gt; -soup” 综合症。
</p>
</body>
</html>
```

在 IE 11.0 中预览效果如图 2-8 所示，<P>标记将文本分成 4 个段落。



图 2-8　段落标记的使用

2.2.2 标题标记<h1>～<h6>

在 HTML 文档中，文本的结构除了以行和段出现之外，还可以作为标题存在。通常一篇文档最基本的结构就是由若干不同级别的标题和正文组成的。

HTML 文档中包含有各种级别的标题，各种级别的标题由<h1>到<h6>元素来定义，<h1>至<h6>标题标记中的字母 h 是英文 headline（标题行）的简称。其中<h1>代表 1 级标题，级别最高，文字也最大，其他标题元素依次递减，<h6>级别最低。

【例 2.6】（实例文件：ch02\2.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>文本段换行</title>
</head>
<body>
<h1>这里是1级标题</h1>
<h2>这里是2级标题</h2>
<h3>这里是3级标题</h3>
<h4>这里是4级标题</h4>
<h5>这里是5级标题</h5>
<h6>这里是6级标题</h6>
</body>
</html>
```

在 IE 11.0 中预览效果如图 2-9 所示。

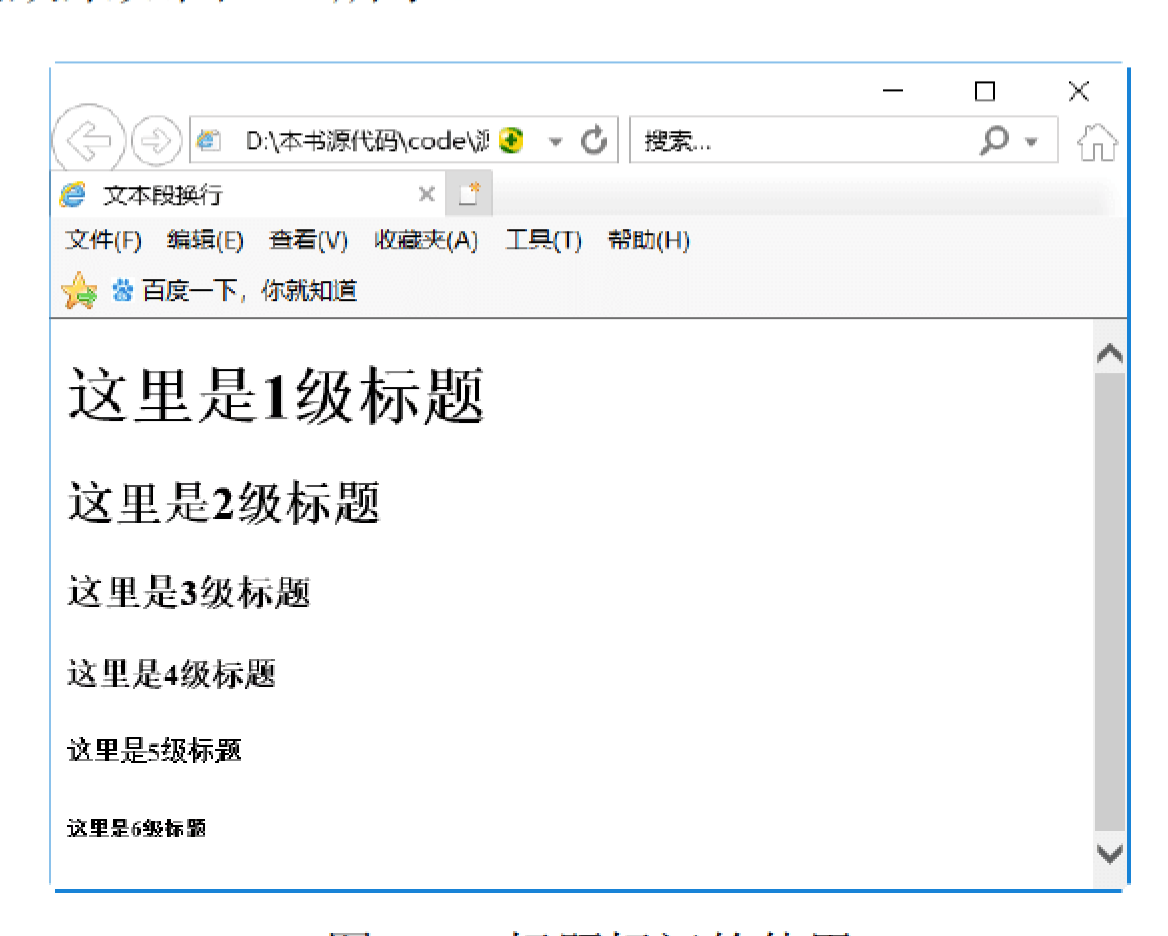


图 2-9 标题标记的使用



作为标题，它们的重要性是有区别的，其中<h1>标题的重要性最高，<h6>的最低。

2.3 文字列表

文字列表可以有序地编排一些信息资源，使其结构化和条理化，并以列表的样式显示出来，以便浏览者能更加快捷地获得相应信息。HTML 中的文字列表如同文字编辑软件 Word 中的项目符号和自动编号。

2.3.1 建立无序列表

无序列表相当于 Word 中的项目符号，无序列表的项目排列没有顺序，只以符号作为分项标识。无序列表使用一对标记
，其中每一个列表项使用
，其结构如下所示：

```
<ul>
  <li>无序列表项</li>
  <li>无序列表项</li>
  <li>无序列表项</li>
  <li>无序列表项</li>
</ul>
```

在无序列表结构中，使用
标记表示这一个无序列表的开始和结束，<li>则表示一个列表项的开始。在一个无序列表中可以包含多个列表项，并且<li>可以省略结束标记。下面使用无序列表实现文本的排列显示。

【例 2.7】（实例文件：ch02\2.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title>嵌套无序列表的使用</title>
</head>

<body>
<h1>网站建设流程</h1>
<ul>
  <li>项目需求</li>
  <li> 系统分析
    <ul>
    <li>网站的定位</li>
    <li>内容收集</li>
    <li>栏目规划</li>
    <li>网站目录结构设计</li>
    <li>网站标志设计</li>
    <li> 网站风格设计</li>
    <li> 网站导航系统设计</li>
    </ul>
   </li>
```

```
        <li> 伪网页草图
         <ul>
         <li> 制作网页草图</li>
         <li>将草图转换为网页</li>
         </ul>
        </li>
        <li> 站点建设</li>
        <li>网页布局</li>
        <li> 网站测试</li>
        <li> 站点的发布与站点管理 </li>
    </ul>
    </body>
    </html>
```

在 IE 11.0 中预览效果如图 2-10 所示。读者会发现，无序列表项中可以嵌套一个列表，如代码中的“系统分析”列表项和“伪网页草图”列表项中都有下级列表，因为在这对
标记间又增加了一对
标记。



图 2-10　无序列表

2.3.2　建立有序列表

有序列表类似于 Word 中的自动编号功能，有序列表的使用方法和无序列表的使用方法基本相同，它使用标记
，每一个列表项使用
。每个项目都有前后顺序之分，通常用数字表示，其结构如下：

```
    <ol>
      <li>第1项</li>
```

```
  <li>第2项</li>
  <li>第3项</li>
</ol>
```

下面实例使用有序列表实现文本的排列显示。

【例 2.8】（实例文件：ch02\2.8.html）

```
<!DOCTYPE html>
<html>
<head>
<title>有序列表的使用</title>
</head>
<body>
<h1>本讲目标</h1>
<ol>
  <li>网页的相关概念 </li>
  <li> 网页与HTML</li>
  <li> Web标准（结构、表现、行为）</li>
  <li> 网页设计与开发的过程  </li>
  <li>与设计相关的技术因素</li>
  <li> HTML简介 </li>
</ol>
</body>
</html>
```

在 IE 11.0 中预览效果如图 2-11 所示。读者可以看到新添加的有序列表。

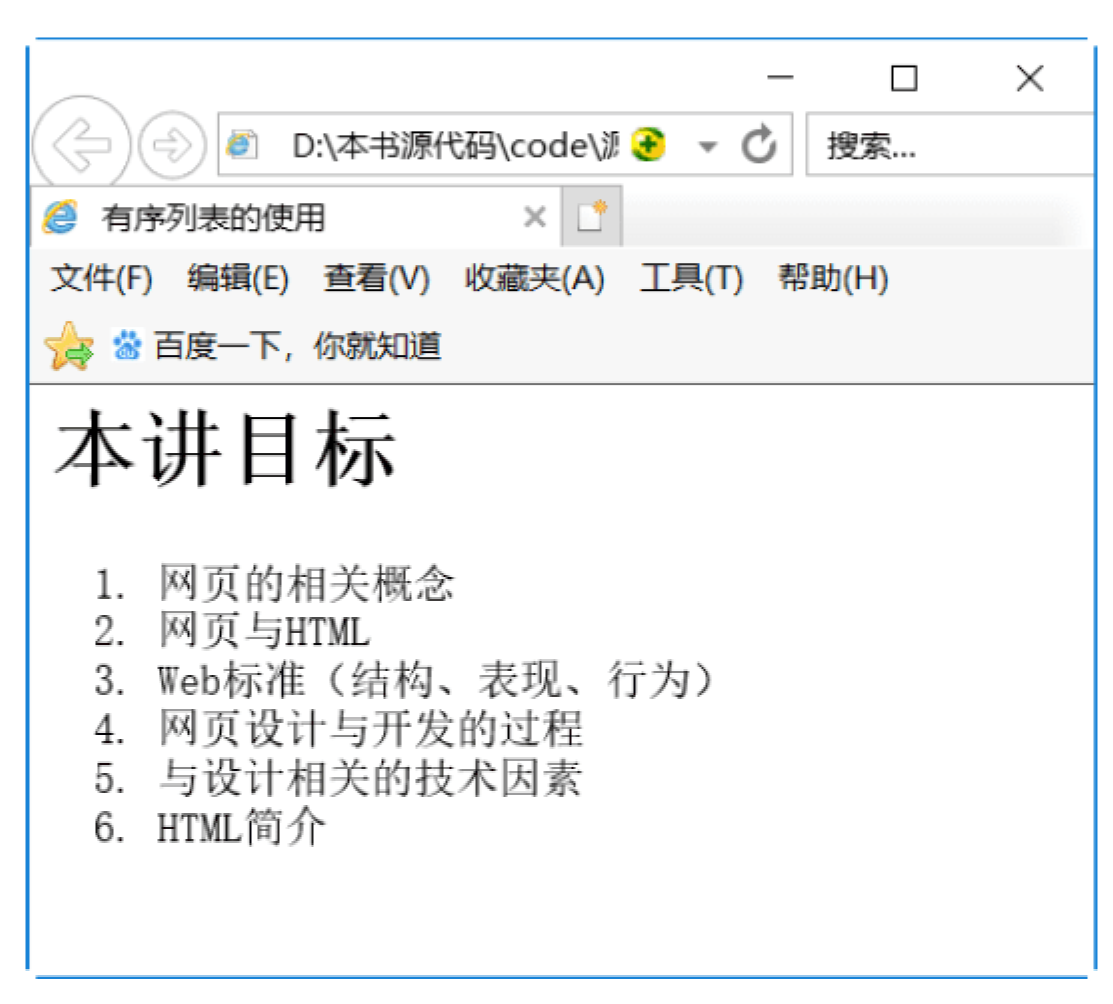


图 2-11　有序列表的效果

2.4　网页中的图像<img>

俗话说“一图胜千言”，图片是网页中不可缺少的元素，巧妙地在网页中使用图片可以为

网页增色不少。网页支持多种图片格式，并且可以对插入的图片设置宽度和高度。

2.4.1 网页中支持的图片格式

图像在网页中具有画龙点睛的作用，它能装饰网页，表达个人的情调和风格。但在网页上加入的图片不宜过多，否则浏览的速度就会受到影响导致用户失去耐心而离开页面。网页中使用的图像可以是 GIF、JPEG、BMP、TIFF、PNG 等格式的图像文件，其中使用最广泛的是 GIF 和 JPEG 两种格式。

GIF 格式是由 Compuserve 公司提出的与设备无关的图像存储标准，也是 Web 上使用最早、应用最广泛的图像格式。GIF 是通过减少组成图像每个像素的存储位数和 LZH 压缩存储技术来降低图像文件大小的。GIF 格式最多只能是 256 色的图像。在低颜色数下，GIF 比 JPEG 装载的更快，可用许多具有同样大小的图像文件组成动画。在 GIF 图像中可指定透明区域，使图像具有非同一般的显示效果。

JPEG 格式是在目前 Internet 中最受欢迎的图像格式。JPEG 可支持多达 16M 颜色，能展现十分丰富生动的图像，还能压缩，但压缩方式是以损失图像质量为代价，压缩比越高图像质量损失越大，图像文件也就越小。

当网页中需要载入一个较大的 GIF 或 JPEG 图像文件时，装载速度会很慢。为改善网页的视觉效果，可在载入时设置为隔行扫描。隔行扫描在显示图像开始看起来非常模糊，接着细节逐渐添加上去直到图像完全显示出来。

GIF 是支持透明、动画的图片格式，但色彩只有 256 色。JPEG 是一种不支持透明和动画的图片格式，但色彩模式比较丰富，可以保留大约 1670 万种颜色。



现在网页中也有很多 PNG 格式的图片。PNG 图片具有不失真、兼有 GIF 和 JPEG 的色彩模式、网络传输速度快、支持透明图像的制作的特点，近年来在网络中很流行。

2.4.2 使用路径

HTML 文档支持文本、图片、声音、视频等媒体格式，但是在这些格式中，除了文本是写在 HTML 中的，其他都是嵌入式的，HTML 文档只记录了这些文件的路径。这些媒体信息能否正确显示，路径至关重要。

路径的作用是定位一个文件的位置。文件的路径可以有两种表述方法：以当前文档为参照物表示文件的位置，即相对路径；以根目录为参照物表示文件的位置，即绝对路径。

为了方便讲述绝对路径和相对路径，可参考如图 2-12 所示的目录结构。

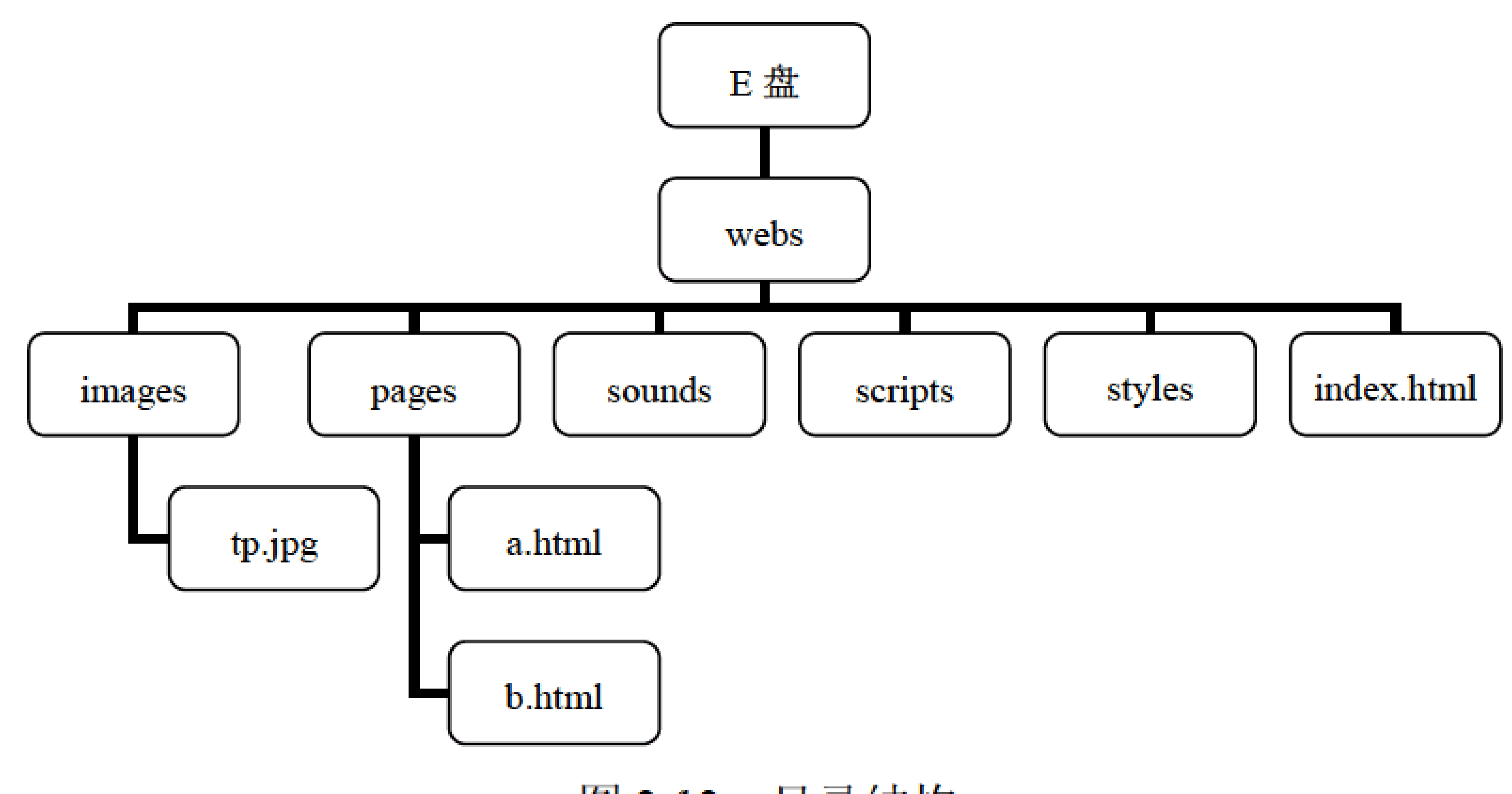


图 2-12　目录结构

1. 绝对路径

例如，在 E 盘 Webs 目录下的 images 中有一个 tp.jpg 图像，它的路径就是 E:\Webs\imags\tp.jpg，像这种完整地描述文件位置的路径就是绝对路径。如果将图片文件 tp.jpg 插入到网页 index.html，绝对路径表示方式如下：

```
E:\Webs\imags\tp.jpg
```

如果使用了绝对路径 E:\Webs\imags\tp.jpg 进行图片链接，那么在本地电脑中将一切正常，因为在 E:\Webs\imags 下确实存在 tp.jpg 这个图片。如果将文档上传到网站服务器之后，就不正常了，因为服务器给你划分的存放空间可能在 E 盘其他目录中，也可能在 D 盘其他目录中，为了保证图片正常显示，必须从 Webs 文件夹开始，放到服务器或其他电脑的 E 盘根目录下。

通过上述讲解读者会发现，当链接本站点内的资源时，使用绝对路径对位置要求非常严格，因此链接本站内的资源不建议采用绝对路径。如果链接其他站点的资源，则必须使用绝对路径。

2. 相对路径

如何使用相对路径设置上述图片呢？所谓相对路径，顾名思义就是以当前位置为参考点，自己相对于与目标的位置。例如，在 index.html 中连接 tp.jpg 就可以使用相对路。index.html 和 tp.jpg 图片的路径根据上述目录结构图可以这样来定位。从 index.html 位置出发，它和 images 属于同级，路径是通的，因此可以定位到 images，images 的下级就是 tp.jpg。使用相对路径表示图片如下：

```
images/tp.jpg
```

使用相对路径，不论将这些文件放到哪里，只要 tp.jpg 和 index.html 文件的相对关系没有变，就不会出错。

在相对路径中，“..”表示上一级目录，“../..”表示上级的上级目录，依此类推。例如，

将 tp.jpg 图片插入到 a.html 文件中，使用相对路径表示如下：

```
../images/tp.jpg
```



细心的读者会发现，路径分隔符使用了“\”和“/”两种，其中“\”表示本地分隔符，“/”表示网络分隔符。因为网站制作好肯定是在网络上运行的，所以要求使用“/”作为路径分隔符。

有的读者可能会有这样的疑惑：一个网站有许多链接，怎么能保证它们的连接都正确？如果调整了一下图片或网页的存储路径，那不是全乱了吗？如何提高工作效率呢？



Dreamweaver 工具的站点管理功能，不但可以将绝对路径自动地转化为相对路径，并且当在站点中改动文件路径时，与这些文件关联的连接路径都会自动更改。

2.4.3　网页中插入图像标记<img>

图像可以美化网页，插入图像使用单标记<img>。<img>标记的属性及描述如表 2-2 所示。

表 2-2　<img>标记的属性及描述

属性	值	描述
alt	text	定义有关图像未加载完成时的提示
title	text	定义鼠标放置在图像上的文本提示
src	URL	要显示的图像的 URL
ismap	URL	把图像定义为服务器端的图像映射
usemap	URL	把图像定义为客户端的图像映射。请参阅 <map> 和 <area> 标记，了解其工作原理
vspace	pixels	定义图像顶部和底部的空白。不推荐使用，请使用 CSS 代替
width	pixels %	设置图像的宽度

1. 图像的源文件 src

src 属性用于指定图片源文件的路径，它是<img>标记必不可少的属性。语法格式如下：

```
<img src="图片路径">
```

图片的路径可以是绝对路径，也可以是相对路径。下面的实例是在网页中插入图片。

【例 2.9】（实例文件：ch02\2.9.html）

```
<!DOCTYPE html>
<html>
<head>
<title>插入图片</title>
</head>
<body>
<img src="images/meishi.jpg">
</body>
</html>
```

在 IE 11.0 中预览效果如图 2-13 所示。



图 2-13　插入图片

2. 设置图像的宽度 width 和高度 height

在 HTML 文档中，还可以设置插入图片的显示大小，一般是按原始尺寸显示，但也可以任意设置显示尺寸。设置图像尺寸分别用属性 width（宽度）和 height（高度）。

【例 2.10】（实例文件：ch02\2.10.html）

```
<!DOCTYPE html>
<html>
<head>
<title>插入图片</title>
</head>
<body>
<!—原始图像、设置宽度为200和设置宽度为200、高度为300-->
<img src="images/meishi.jpg">
<img src="images/meishi.jpg" width="200">
<img src="images/meishi.jpg" width="200" height="300">
```

```
</body>
</html>
```

在 IE 11.0 中预览效果，如图 2-14 所示。



图 2-14　设置图片的宽度和高度

由图 2-14 可以看到，图片的显示尺寸是由 width（宽度）和 height（高度）控制的。当只为图片设置一个尺寸属性时，另外一个尺寸就以图片原始的长宽比例来显示。图片的尺寸单位可以选择百分比或数值。百分比为相对尺寸，数值是绝对尺寸。



因为网页中插入的图像都是位图，所以放大尺寸，图像会出现马赛克，变得模糊。在 Windows 中查看图片的尺寸，只需要找到图像文件，把鼠标指针移动到图像上，停留几秒后，就会出现一个提示框，说明图像文件的尺寸。尺寸后显示的数字，代表图像的宽度和高度，如 256 × 256。

3. 设置图像的提示文字 alt

在浏览网页时，图像提示文字的作用有两个：一是如果图像下载完成，将鼠标指针放在该图像上，鼠标指针旁边就会出现提示文字；二是如果图像没有成功下载，在图像的位置上就会显示提示文字。

随着互联网技术的发展，网速已经不是制约因素，因此一般都能成功下载图像。现在，alt 还有另外一个作用，在百度、google 等大搜索引擎中，搜索图片不如文字方便，如果给图片添加适当提示，就可以方便搜索引擎的检索。

下面实例为图片添加提示文字效果。

【例 2.11】（实例文件：ch02\2.11.html）

```
<!DOCTYPE html>
<html>
```

```
    <head>
    <title>图片文字提示</title>
    </head>
    <body>
    <!—添加提示文字效果-->
    <img src="images/meishi.jpg" alt="未加载完成时显示的替代文字" title="鼠标放上去显示的文字">
    </body>
    </html>
```

在 IE 11.0 中预览效果如图 2-15 所示。用户将鼠标放在图片上，即可看到提示文字。



图 2-15　图片文字提示

注意，在火狐浏览器中不支持该功能。

2.5　综合实例——图文并茂房屋装饰装修网页

本章讲述了网页组成元素中最常用的文本和图片。本综合实例的目的是创建一个由文本和图片构成的房屋装饰效果网页（如图 2-16 所示），具体操作步骤如下：

01 在 Dreamweaver CC 中新建 HTML 文档，并修改成 HTML5 标准，代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>房屋装饰装修效果图</title>
</head>
<body>
</body>
</html>
```

02 在 body 部分增加如下 HTML 代码，保存页面。

```
<p> <img src="images/xiyatu.jpg" width="300" height="200"/><br/>
西雅图原生态公寓室内设计</p>
<hr>   <!—此标签将显示为一条水平线。-->
<p> <img src="images/stadshem.jpg" width="300" height="200"/><br/>
Stadshem小户型公寓设计（带阁楼）</p>
<hr>
<p> <img src="images/qingxinhuoli.jpg" width="300" height="200"/><br/>
清新活力家居</p>
<hr>
<p> <img src="images/renwen.jpg" width="300" height="200"/><br/>
人文简约悠然家居</p>
<hr>
```

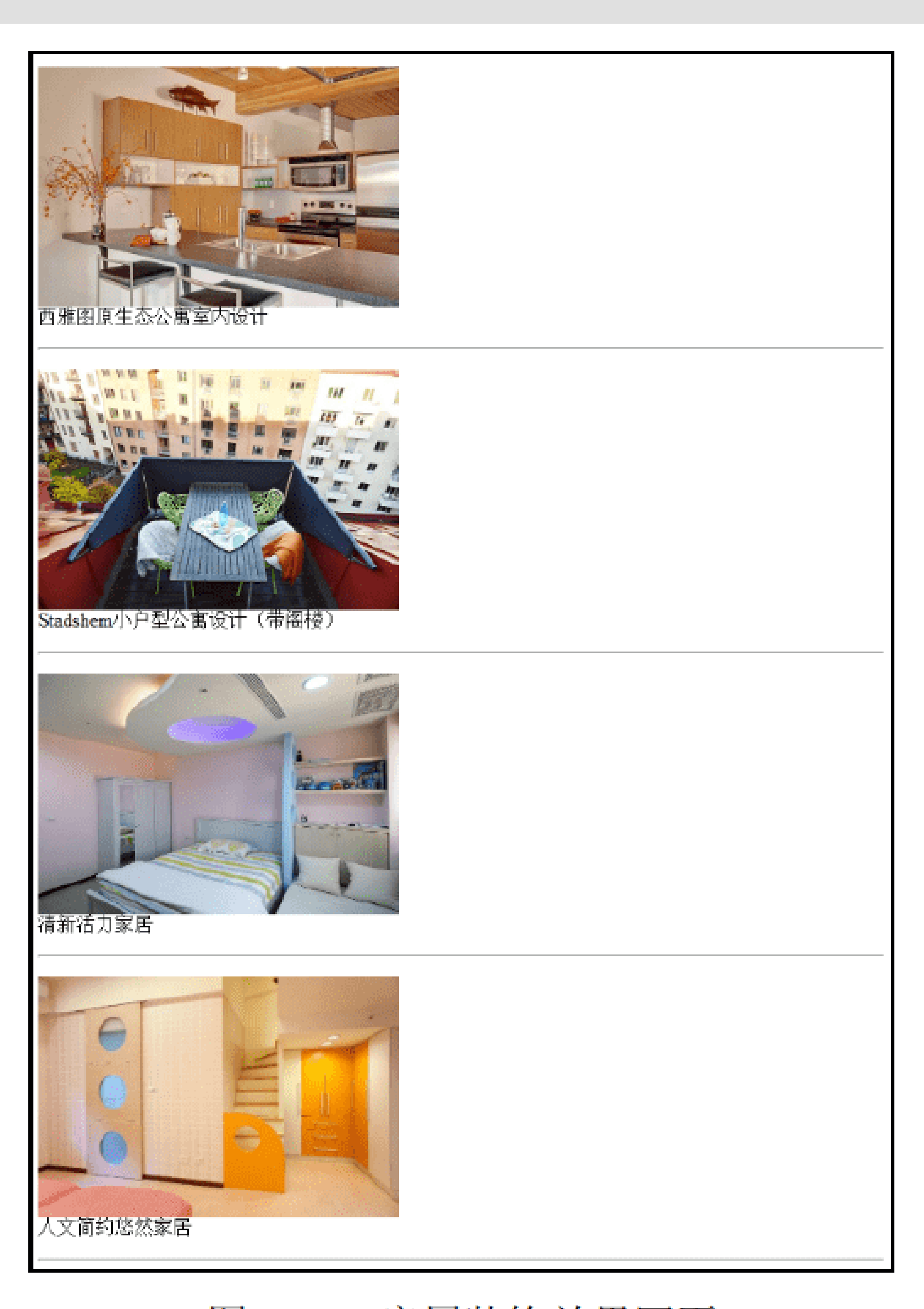


图 2-16　房屋装饰效果网页



<hr>标记的作用是定义内容中的主题变化，并显示为一条水平线，在 HTML5 中它没有任何属性。

另外，快速插入图片及设置相关属性可以借助 Dreamweaver CC 的插入功能，或者按下组合键 Ctrl+Alt+I。

2.6 专家解惑

问题 1：换行标记和段落标记的区别？

答：换行标记是单标记，不能写结束标记。段落标记是双标记，可以省略结束标记也可以不省略。默认情况下，段落之间的距离和段落内部的行间距是不同的，段落间距比较大，行间距比较小。HTML 是无法调整段落间距和行间距的，如果需要调整它们，就必须使用 CSS。在 Dreamweaver CC 的设计视图下，按 Enter 键可以快速换段，按 Shift+ Enter 组合键可以快速换行。

问题 2：无序列表<ul>标记的作用？

答：无序列表标记主要用于条理化和结构化文本信息。在实际开发中，无序列表在制作导航菜单时使用广泛。导航菜单的结构一般都使用无序列表实现。

问题 3：在浏览器中，图片无法显示。

答：图片在网页中属于嵌入对象，并不是图片保存在网页中，网页只是保存了指向图片的路径。浏览器在解释 HTML 文件时，会按指定的路径去寻找图片，如果在指定的位置不存在图片，就无法正常显示。为了保证图片的正常显示，制作网页时需要注意以下几点。

（1）图片格式一定是网页支持的。

（2）图片的路径一定要正确，并且图片文件扩展名不能省略。

（3）HTML 文件位置发生改变时，图片一定要随着改变，即图片位置和 HTML 文件位置始终保持相对一致。

第3章　用HTML5建立超链接

HTML 文件中最重要的应用之一就是超链接。超链接是一个网站的灵魂，Web 上的网页是互相链接的，单击被称为超链接的文本或图形就可以链接到其他页面。

3.1　URL 的概念

URL 为“Uniform Resource Locator”的缩写，通常翻译为“统一资源定位器”，也就是人们通常说的“网址”，它用于指定 Internet 上的资源位置。

3.1.1　URL 的格式

网络中的计算机是通过 IP 地址区分的，如果需要访问网络中某台计算机中的资源，首先要定位到这台计算机。IP 地址由 32 位二进制代码（即 32 个 0/1）组成，数字之间没有意义，且不容易记忆。为了方便记忆，现在计算机一般采用域名的方式来寻址，即在网络上使用一组有意义字符组成的地址代替 IP 地址来访问网络资源。

URL 由 4 个部分组成，即“协议”“主机名”“文件夹名”“文件名”，如图 3-1 所示。



图 3-1　URL 组成

互联网中有各种各样的应用，如 Web 服务、FTP 服务等。每种服务应用都得对应有协议，通过浏览器浏览网页的协议都是 HTTP 协议（超文本传输协议），因此通常网页的地址都以“http://”开头。

“www.baidu.com”为主机名，表示文件存在于哪台服务器，主机名可以通过 IP 地址或域名来表示。

确定到主机后，还需要说明文件存在于这台服务器的哪个文件夹中，这里文件夹可以分为多个层级。

确定文件夹后，就要定位到文件，即要显示哪个文件，网页文件通常是以“.html”或“htm”为扩展名。

3.1.2 URL 的类型

在第 2 章讲解网页中使用的图像时，已经介绍了“路径”的概念。对于超链接来说，路径的概念同样存在。

超链接的 URL 可以分为绝对 URL 和相对 URL 两种类型。

（1）绝对 URL 一般用于访问非同一台服务器上的资源。

（2）相对 URL 是指访问同一台服务器上相同文件夹或不同文件夹中的资源。如果访问相同文件夹中的文件，只需要写文件名；如果访问不同文件夹中资源，URL 以服务器的根目录为起点，指明文件的相对关系，由文件夹名和文件名两部分构成。

下面实例使用绝对 URL 和相对 URL 实现超链接。

【例 3.1】（实例文件：ch03\3.1.html）

```
<!DOCTYPE html>
<html>
<head>
<title>绝对URL和相对URL</title>
</head>
<body>
<!--使用绝对URL-->
单击<a href="http://www.WebDesign.com/index.html">绝对URL</a>链接到
WebDesign网站首页<br/>
<!--使用相对URL-->
单击<a href="02.html">相同文件夹的URL</a>链接到相同文件夹中的第2个页面<br/>
单击<a href="../pages/03.html">不同文件夹的URL</a>链接到不同文件夹中的第3个页面
</body>
</html>
```

在上述代码中，第 1 个链接使用的是绝对 URL；第 2 个链接使用的是服务器相对 URL，也就是链接到文档所在服务器的根目录下的 02.html 文件；第 3 个链接使用的是文档相对 URL，即原文档所在文件夹的父文件夹下面的 pages 文件夹中的 03.html 文件。

在 IE 11.0 中预览网页效果如图 3-2 所示。

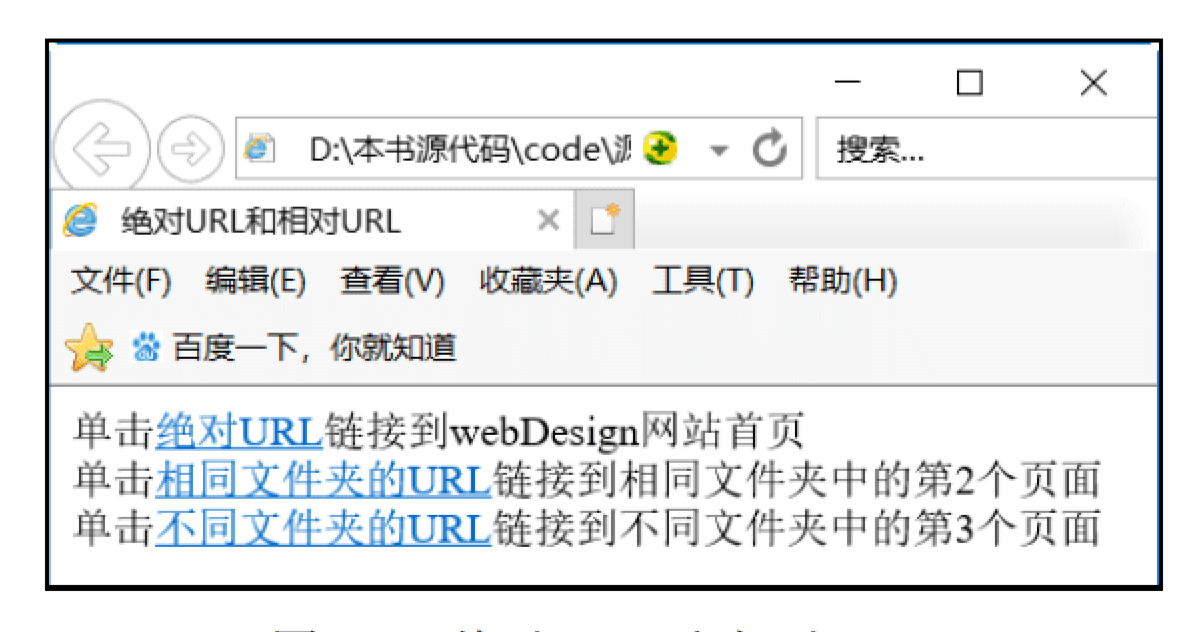


图 3-2 绝对 URL 和相对 URL

3.2　超链接标记<a>

超链接是指当鼠标单击一些文本、图片或其他网页元素时，浏览器会根据其指示载入一个新的页面或跳转到页面的其他位置。超级链接除了可链接文本外，还可链接各种多媒体，如声音、图像、动画等，通过它们可享受丰富多彩的多媒体世界。

建立超链接所使用的 HTML 标记为<a>。超链接最重要的两个要素是超链接指向的目标地址和设置为超链接的网页元素。基本的超链接结构如下：

```
<a href=URL>网页元素</a>
```

3.2.1　设置文本和图片的超链接

设置超链接的网页元素通常使用文本和图片。文本超链接和图片超链接通过<a>标记实现，将文本或图片放在<a>开始标记和结束标记之间即可建立超链接。

下面实例将实现文本和图片的超链接。

【例 3.2】（实例文件：ch03\3.2.html）

```
<!DOCTYPE html>
<html>
<head>
<title>文本和图片超链接</title>
</head>
<body>
<a href="a.html"><img src="images/0371.gif"></a>
<a href="b.html">公司简介</a>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 3-3 所示。用鼠标单击图片或文本即可实现链接跳转的效果。

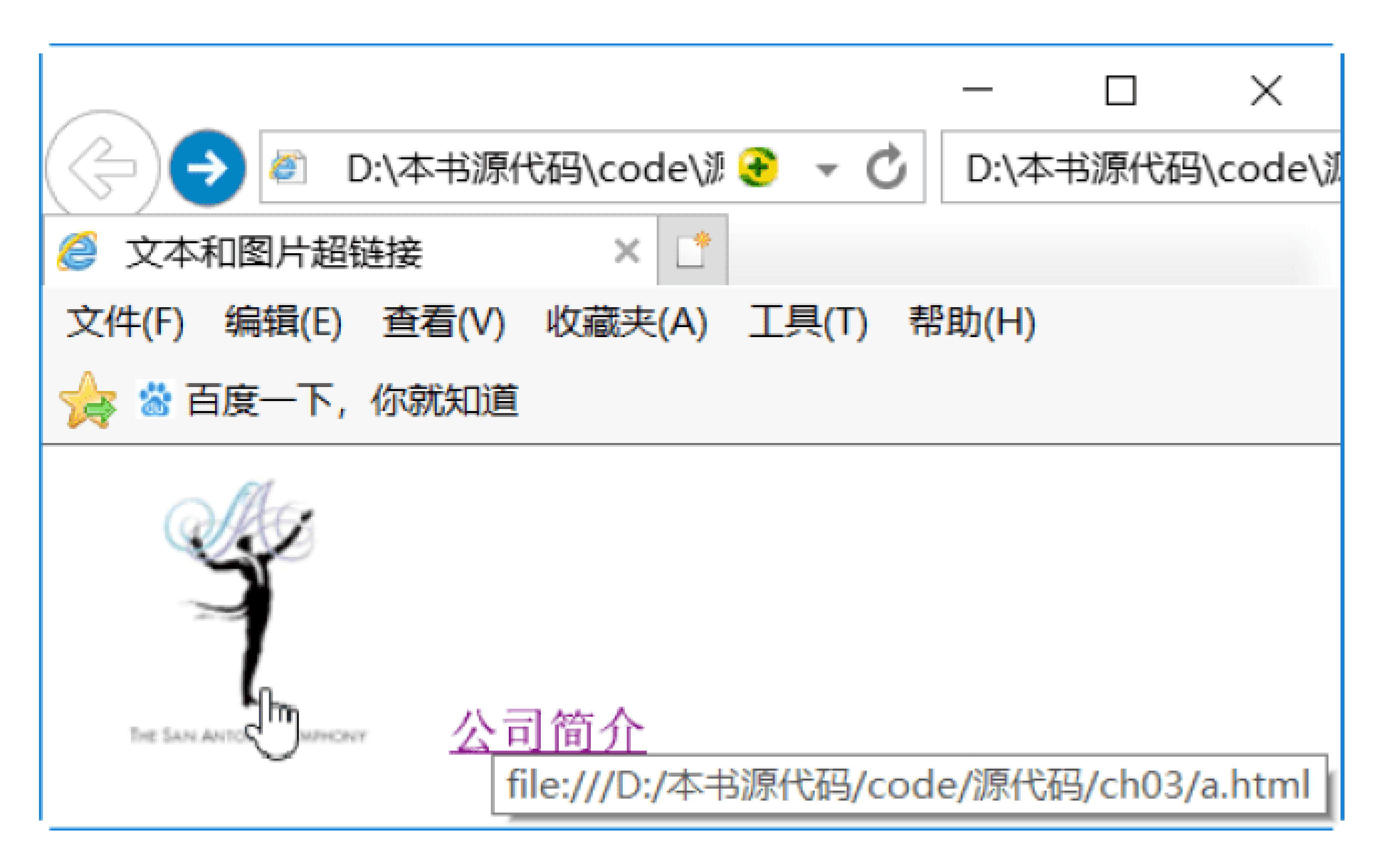


图 3-3　文本和图片链接效果

默认情况下，为文本添加超链接，文本会自动增加下画线，并且文本颜色变为蓝色，单击过的超链接，文本会变成暗红色。图片增加超链接以后，浏览器会自动给图片加一个粗边框。图片和文本超链接的下画线需要借助 CSS 样式完成，这里不作介绍，详见 CSS 部分。

3.2.2 超链接指向的目标类型

通过上面的讲解，读者会发现超链接的目标对象都是.html 类型的文件。超链接不但可以链接到各种类型（如图片文件、声音文件、视频文件、word 等）的文件，还可以链接到其他网站、ftp 服务器、电子邮件等。

1. 链接到各种类型的文件

超链接<a>标记的 href 属性指向链接的目标，目标可以是各种类型的文件。如果是浏览器能够识别的类型，就会直接在浏览器中显示；如果是浏览器不能识别的类型，在 IE 浏览器中会弹出文件下载对话框，如图 3-4 所示。



图 3-4　IE 中的文件下载对话框

【例 3.3】（实例文件：ch03\3.3.html）

```
<!DOCTYPE html>
<html>
<head>
<title>链接各种类型文件</title>
</head>
<body>
<p><a href="a.html">链接html文件</a></p>
<p><a href="coffe.jpg">链接图片</a></p>
<p><a href="2.doc">链接word文档</a></p>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 3-5 所示。实现链接到 HTML 文件、图片和 Word 文档。

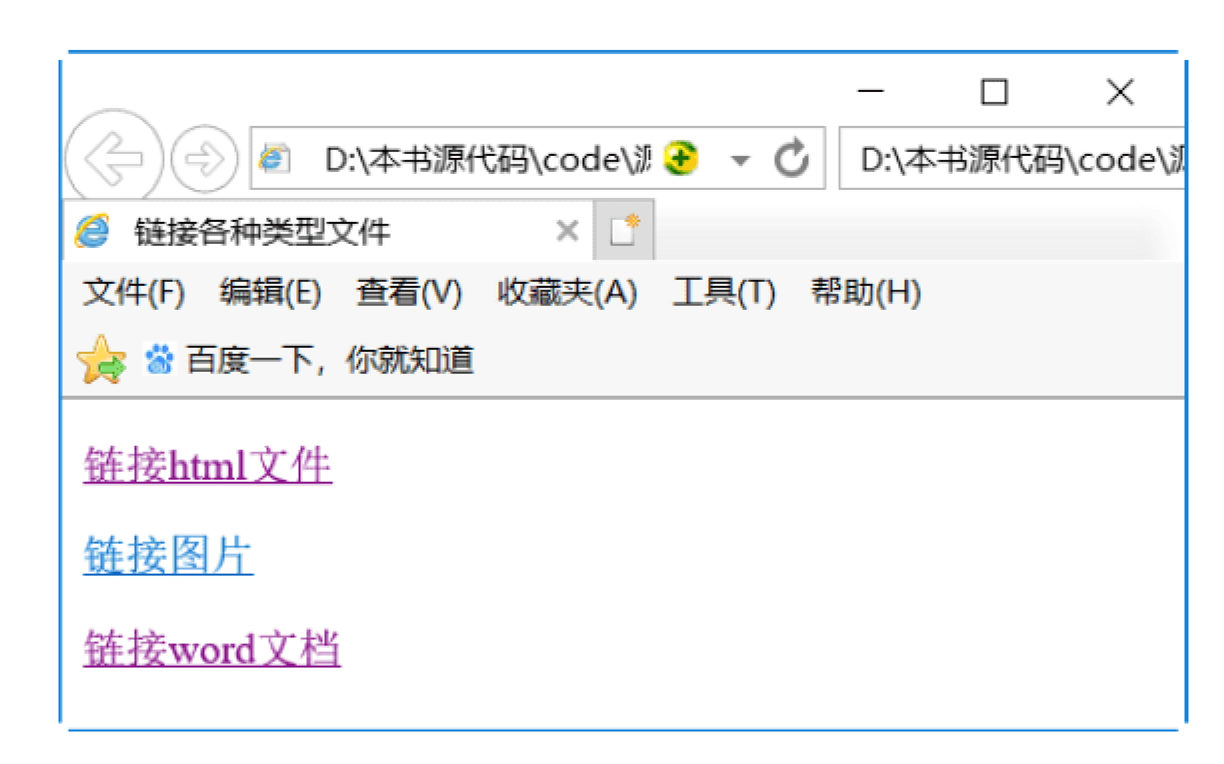


图 3-5　各种类型的链接

2. 链接到其他网站或 FTP 服务器

在网页中，友情链接也是推广网站的一种方式。下列代码实现了链接到其他网站或 FTP 服务器的功能。

```
<a href="http://www.baidu.com">链接百度</a>
<a href="ftp://172.16.1.254">链接到ftp服务器</a>
```



这里 FTP 服务器用的是 IP 地址。为了保证代码的正确运行，请读者填写有效的 FTP 服务器地址。

3. 设置电子邮件链接

在某些网页中，当访问者单击某个链接以后，会自动打开电子邮件客户端软件（如 Outlook 或 Foxmail 等）向某个特定的 E-mail 地址发送邮件，这个链接就是电子邮件链接。电子邮件链接的格式如下：

```
<a href="mailto:电子邮件地址" >网页元素</a>
```

【例 3.4】（实例文件：ch03\3.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>电子邮件链接</title>
</head>
<body>
<img src="images/logo.gif" width="119" height="49">  [免费注册][登录]
<a href="mailto:kfdzsj@126.com">站长信箱</a>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 3-6 所示，实现了电子邮件链接。

当读者单击【站长信箱】链接时，会自动弹出电子邮件客户端窗口以编写电子邮件，如图 3-7 所示。

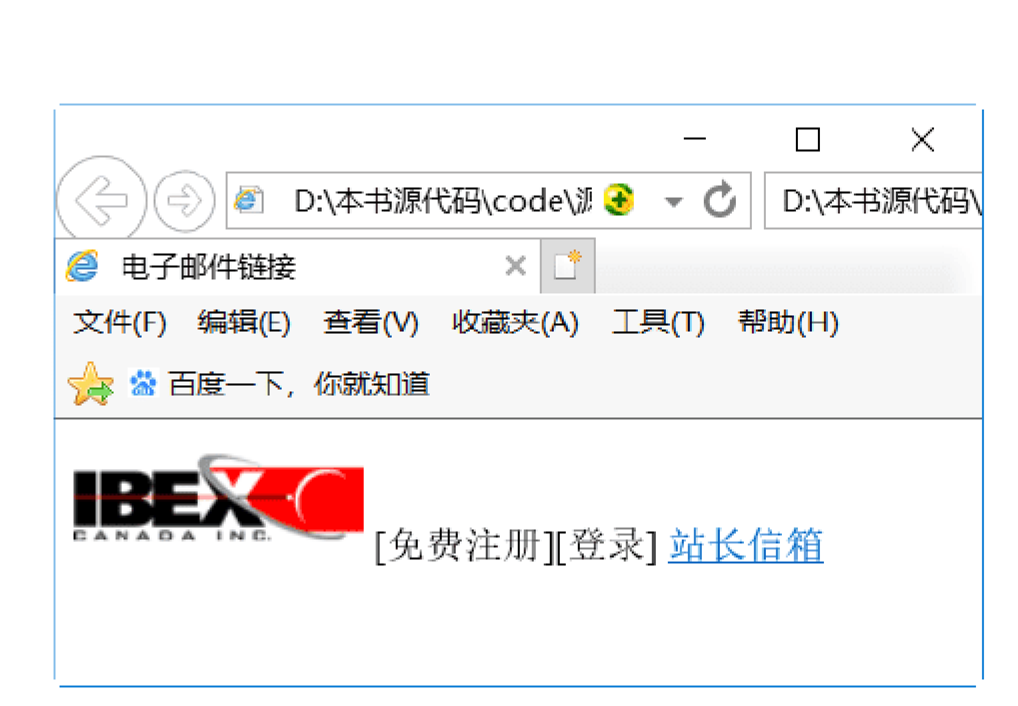



图 3-6 链接到电子邮件

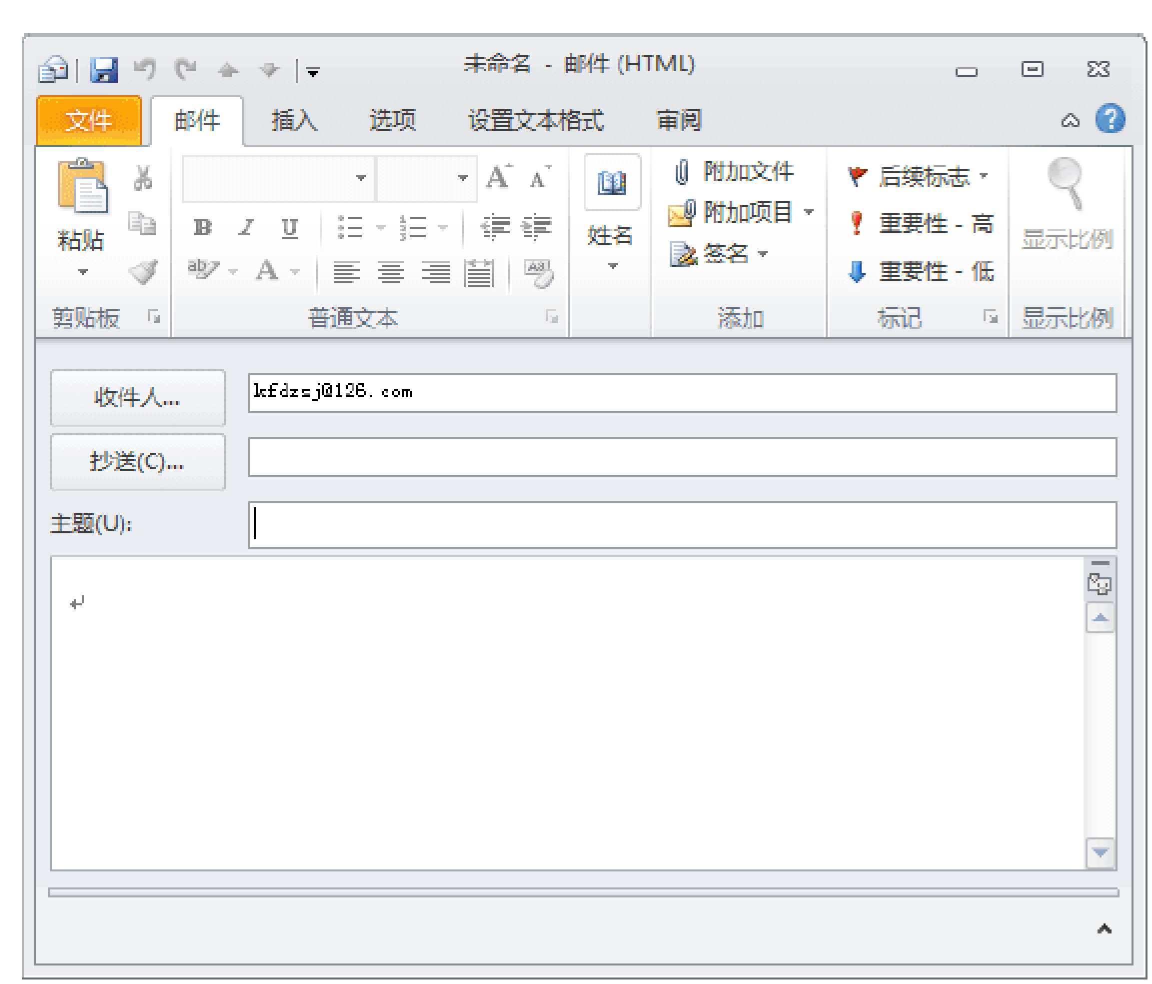


图 3-7 电子邮件客户端窗口

3.2.3 设置以新窗口显示超链接页面

默认情况下，当单击超链接时，目标页面会在当前窗口中显示并替换掉当前页面的内容。如果要实现在单击某个超链接后在新的浏览器窗口中显示目标页面，就需要使用<a>标记的 target 属性。target 属性取值有 4 个：_blank、_self、_top、_parent。由于 HTML5 不再支持框，所以_top、_parent 这两个取值不常用。本小节仅为读者讲解_blank、_self 值。其中，_blank 表示在新窗口中显示超链接页面；_self 表示在当前窗口中显示超链接页面。当省略 target 属性时，默认取值为_self。

【例 3.5】（实例文件：ch03\3.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>以新窗口方式打开</title>
</head>
<body>
<a href="a.html target="_blank">新窗口</a>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 3-8 所示。

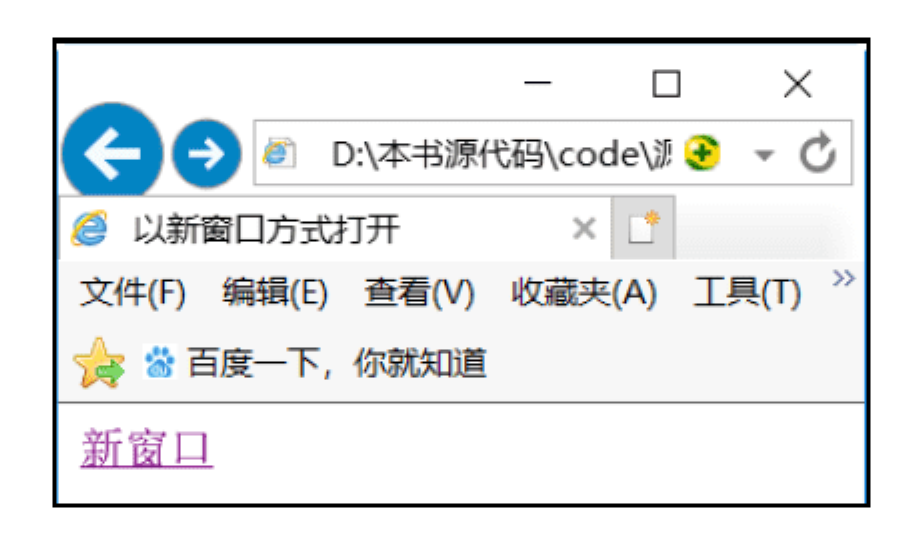


图 3-8　新窗口页面

3.3　创建热点区域

在浏览网页时读者会发现，当单击一张图片的不同区域时会显示不同的链接内容，这就是图片的热点区域。所谓图片的热点区域就是将一个图片划分成若干个链接区域。访问者单击不同的区域会链接到不同的目标页面。

在 HTML 中，可以为图片创建 3 种类型的热点区域：矩形、圆形和多边形。创建热点区域使用标记<map>和<area>，语法格式如下：

```
<img src="图片地址" usemap="#名称">
<map id="#名称">
   <area shape="rect" coords="10,10,100,100" href="#">
   <area shape="circle" coords="120,120,50" href="#">
   <area shape="poly" coords="78,13,81,14,53,32,86,38" href="#">
</map>
```

在上面的语法格式中，需要注意以下几点：

（1）要想建立图片热点区域，必须先插入图片。注意，图片必须增加 usemap 属性，说明该图像是热区映射图像，属性值必须以“#”开头，如#pic。那么上面一行代码可以修改为：<img src="图片地址" usemap="#pic">。

（2）<map>标记只有一个属性 id，其作用是为区域命名，属性值必须与<img>标记的 usemap 属性值相同，修改上述代码为<map id="#pic">

（3）<area>标记主要是定义热点区域的形状及超链接，它有 3 个相应的属性。

- shape 属性，控件划分区域的形状，其取值有 3 个，分别是 rect（矩形）、circle（圆形）和 poly（多边形）。
  - 如果 shape 属性取值为 rect，那么 coords 的设置值分别为矩形的左上角 x、y 坐标点和右下角 x、y 坐标点，单位为像素。
  - 如果 shape 属性取值为 circle，那么 coords 的设置值分别为圆形圆心 x、y 坐标点和半径值，单位为像素。

◆ 如果 shape 属性取值为 poly，那么 coords 的设置值分别为多边形在各个点 x、y 坐标，单位为像素。

- coords 属性，控制区域的划分坐标。
- href 属性是为区域设置超链接的目标，设置值为“#”时，表示为空链接。

3.4 浮动框架 iframe

HTML5 中已经不支持 frameset 框架，但是它仍然支持 iframe 浮动框架的使用。浮动框架可以自由控制窗口大小，可以配合表格随意地在网页的任何位置插入窗口。实际上就是在窗口中再创建一个窗口。

使用 iframe 创建浮动框架的格式如下：

```
<iframe src="链接对象" >
```

其中，src 表示浮动框架中显示对象的路径，可以是绝对路径，也可以是相对路径。例如，下面的代码是在浮动框架中显示到百度网站。

【例 3.6】（实例文件：ch03\3.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>浮动框架中显示百度网站</title>
</head>
<body>
<iframe src="http://www.baidu.com"></iframe>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 3-9 所示。



图 3-9 浮动框架效果

从预览结果可见，浮动框架在页面中又创建了一个窗口。默认情况下，浮动框架的宽度和高度为 220×120 像素，如果需要调整浮动框架尺寸，请使用 CSS 样式。修改上述浮动框架尺寸，需在<head></head>标记中间增加如下 CSS 代码。

```
<style>
iframe{
    width:1200px;     /*设置浮动框架的宽度*/
    height:800px;   /*设置浮动框架的高度*/
    border:none;     /*设置无边框效果*/
}
</style>
```

在 IE 11.0 中预览网页效果如图 3-10 所示。



图 3-10　修改宽度和高度的浮动框架



在 HTML5 中，iframe 仅支持 src 属性，再无其他属性。

3.5　综合实例——用 Dreamweaver 精确定位热点区域

上面讲述了 HTML 创建热点区域的方法，其中最让读者头痛的地方就是坐标点的定位。对于简单的形状还可以，如果边数较多且形状复杂，那么确定坐标点的工程量就很大，因此不建议使用 HTML 代码去完成。这里将为读者介绍一个快速且能精确定位热点区域的方法，使用 Dreamweaver CC 可以很方便地实现这个功能。

Dreamweaver CC 创建图片热点区域的具体操作步骤如下：

01 创建一个 HTML 文档，插入一个图片文件，如图 3-11 所示。



图 3-11　插入图片

02 选择图片，在 Dreamweaver CC 中打开【属性】面板，面板左下角有 3 个蓝色图标按钮，依次代表矩形、圆形和多边形热点区域。单击左边的【矩形热点】工具图标，如图 3-12 所示。

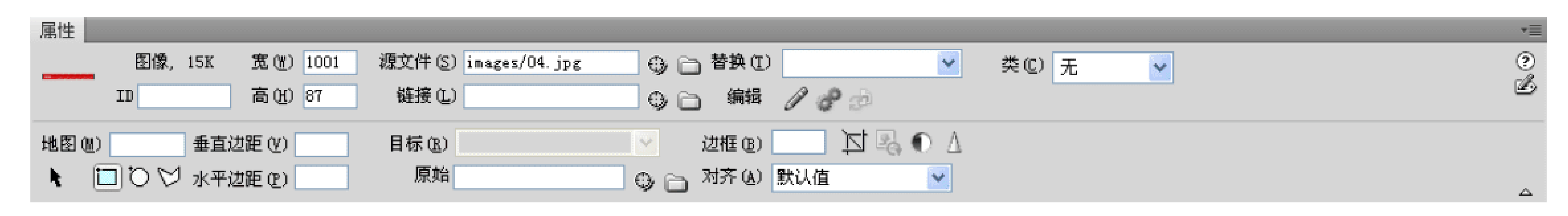


图 3-12　Dreamweaver CC 中的【属性】面板

03 将鼠标指针移动到图片上，以【创意信息平台】栏中的矩形大小为准，按住鼠标左键从左上方向右下方拖曳，得到矩形区域，如图 3-13 所示。

04 绘制出来的热区呈现出半透明状态，效果如图 3-14 所示。



图 3-13　绘制矩形热点区域

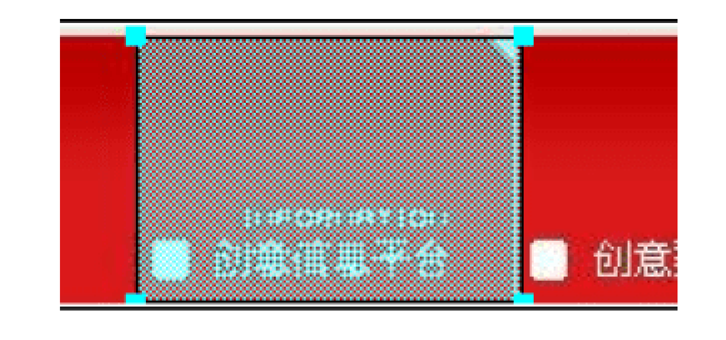


图 3-14　完成矩形热点区域的绘制

05 如果绘制出来的矩形热区有误差，可以通过【属性】面板中的【指针热点】工具进行编辑，如图 3-15 所示。



图 3-15　“指针热点”工具

06 完成上述操作之后，保持矩形热区被选中状态，然后在【属性】面板中的【链接】文本框中输入该热点区域链接对应的跳转目标页面。

07 在【目标】下拉列表框中有 4 个选项，它们决定着链接页面的弹出方式，这里如果选择了【_blank】，那么矩形热区的链接页面将在新的窗口中弹出。如果【目标】选项保持空白，

就表示仍在原来的浏览器窗口中显示链接的目标页面。这样，矩形热点区域就设置好了。

08 接下来继续为其他菜单项创建矩形热点区域。操作方法请参阅上面步骤，完成后的效果如图 3-16 所示。

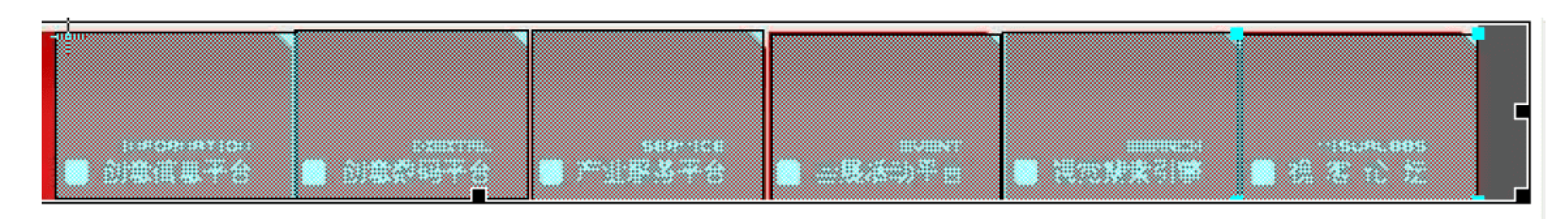


图 3-16　为其他菜单项创建矩形热点区域

09 完成后保存并预览页面。可以发现，凡是绘制了热点的区域，鼠标指针移上去时就会变成手形，单击就会跳转到相应的页面。

10 到此为止，使用热点区域制作网站的导航就完成了。此时页面相应的 HTML 源代码如下：

```
    <!DOCTYPE html>
    <html>
    <head>
    <title>创建热点区域</title>
    </head>
    <body>
    <img src="images/04.jpg" width="1001" height="87" border="0"
usemap="#Map">
    <map name="Map">
      <area shape="rect" coords="298,5,414,85" href="#">
      <area shape="rect" coords="412,4,524,85" href="#">
      <area shape="rect" coords="525,4,636,88" href="#">
      <area shape="rect" coords="639,6,749,86" href="#">
      <area shape="rect" coords="749,5,864,88" href="#">
      <area shape="rect" coords="861,6,976,86" href="#">
    </map>
    </body>
    </html>
```

可以看到，Dreamweaver CC 自动生成的 HTML 代码结构和前面介绍的是一样的，所有的坐标都自动计算出来了，这正是网页制作工具的快捷之处。使用这些工具本质上与手工编写 HTML 代码没有区别，只是使用这些工具可以提高工作效率。



本书所讲述手工编写 HTML 代码的方法，在 Dreamweaver CC 工具中几乎都有对应的操作，请读者自行研究，以提高编写 HTML 代码效率。但是请读者注意，使用网页制作工具前，一定要明白这些 HTML 标记的作用。因为一个专业的网页设计师必须具备 HTML 方面的知识，不然再强大的工具也只能是无根之树，无源之泉。

3.6　专家解惑

1. 在创建超链接时，使用绝对 URL 还是相对 URL？

答：在创建超链接时，如果要链接的是另外一个网站中的资源，就需要使用完整的绝对 URL；如果在网页中创建内部链接，一般使用相对当前文档或站点根文件夹的相对 URL。

2. 链接增多后，网站如何设置目录结构以方便维护？

答：当一个网站的网页数量增加到一定程度以后，网站的管理与维护将变得非常烦琐，因此掌握一些网站管理与维护的技术是非常实用的，可以节省很多时间。建立适合的网站文件存储结构，可以方便网站的管理与维护。通常使用的 3 种网站文件组织结构方案及文件管理遵循的原则如下：

（1）按照文件的类型进行分类管理。将不同类型的文件放在不同的文件夹中，这种存储方法适合于中小型的网站，通过文件的类型对文件进行管理。

（2）按照主题对文件进行分类。网站的页面按照不同的主题进行分类存储。同一主题的所有文件存放在一个文件夹中，再进一步细分文件的类型。这种方案适用于页面与文件数量众多、信息量大的静态网站。

（3）对文件类型进行进一步细分存储管理。这种方案是第一种存储方案的深化，将页面进一步细分后进行分类存储管理，适用于文件类型复杂、包含各种文件的多媒体动态网站。

第4章　用HTML5创建表格

在 HTML 中，表格不但可以清晰地显示数据，而且还可以用于页面布局。目前表格布局已经被抛弃，本章只为读者介绍表格如何显示数据。

4.1　表格基本结构及操作

HTML 中的表格类似于 Word 软件中的表格，尤其在使用网页制作工具时，基本操作也很相似。HTML 制作表格的原理是使用相关标记定义完成，如表格对象<table>、行对象<tr>、单元格对象<td>，其中单元格的合并在表格操作中应用广泛。

4.1.1　表格基本结构

使用表格显示数据，可以更直观和清晰。在 HTML 文档中表格主要用于显示数据，虽然可以使用表格布局，但是不建议使用，它有很多弊端。表格一般由行、列和单元格组成，如图 4-1 所示。

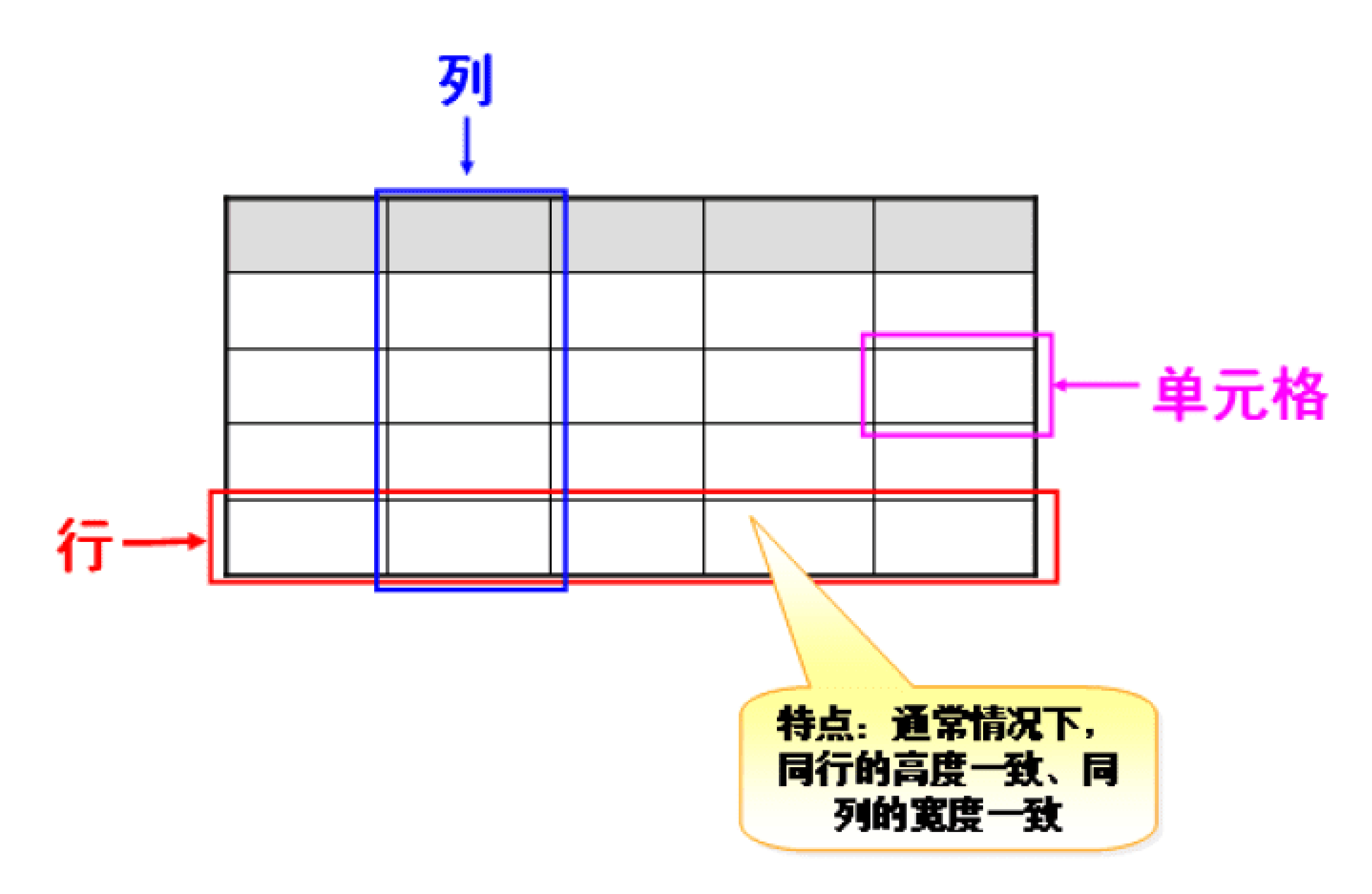


图 4-1　表格的组成

<table>标记用于标识一个表格对象的开始，</table>标记用于标识一个表格对象的结束。一个表格中，只允许出现一对<table>标记。在 HTML5 中不再支持它的任何属性。

<tr>标记用于标识表格一行的开始，</tr>标记用于标识表格一行的结束。表格内有多少对<tr></tr>标记，就表示表格中有多少行。在 HTML5 中不再支持它的任何属性。

<td>标记用于标识表格某行中的一个单元格开始，</td>标记用于标识表格某行中的一个单元格结束。<td></td>标记书写在<tr></tr>标记内，一对<tr></tr>标记内有多少对<td></td>标记，就表示该行有多少个单元格。在 HTML5 中仅有 colspan 和 rowspan 两个属性，详见 4.1.2 小节。

最基本的表格必须包含一对<table></table>标记、一对或几对<tr></tr>标记及一对或几对<td></td>标记。一对<table></table>标记定义一个表格，一对<tr></tr>标记定义一行，一对<td></td>标记定义一个单元格。

例如定义一个 4 行 3 列的表格。

【例 4.1】（实例文件：ch04\4.1.html）

```
<!DOCTYPE html>
<html>
<head>
<title>表格基本结构</title>
</head>
<body>
<table>
  <tr>
  <td>A1</td>
  <td>B1</td>
  <td>C1</td>
  </tr>
  <tr>
  <td>A2</td>
  <td>B2</td>
  <td>C2</td>
  </tr>
  <tr>
  <td>A3</td>
  <td>B3</td>
  <td>C3</td>
  </tr>
  <tr>
  <td>A4</td>
  <td>B4</td>
  <td>C4</td>
  </tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 4-2 所示。

从预览图中读者会发现，表格没有边框，行高及列宽也无法控制。上面提到 HTML5 中除了<td>标记提供两个单元格合并属性之外，<table>和<tr>标记已没有任何属性。那么如何修饰表格呢？表格的所有外观设置都需要通过 CSS 样式完成，详见 CSS 章节部分。

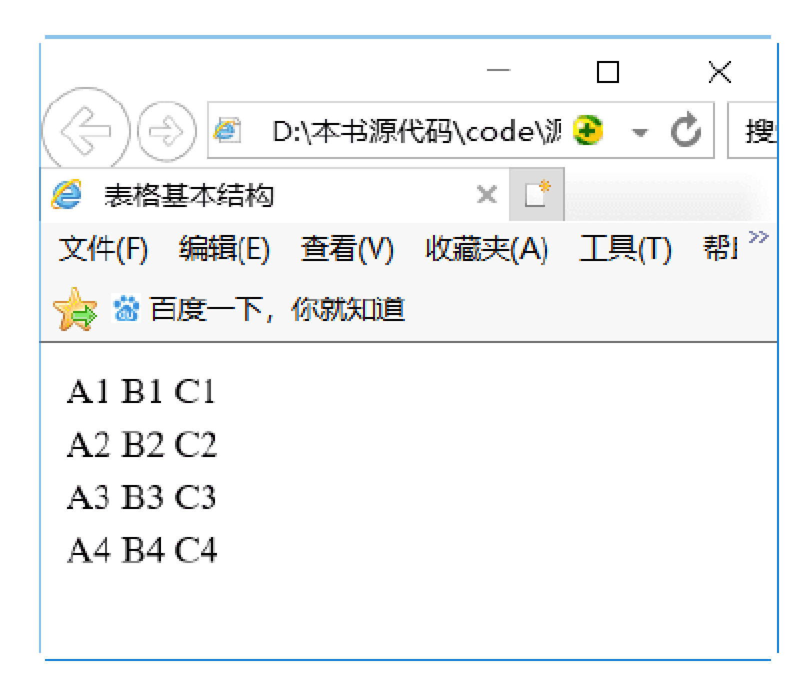


图 4-2　表格基本结构

4.1.2　合并单元格

在实际应用中，并非所有表格都是规范的几行几列，有时需要将某些单元格进行合并，以符合某种内容上的需要。在 HTML 中合并的方向有两种：一种是上下合并；另一种是左右合并，这两种合并方式只需要使用<td>标记的两个属性即可。

1. 用 colspan 属性合并左右单元格

左右单元格的合并需要使用<td>标记的 colspan 属性完成，格式如下：

```
<td colspan="数值">单元格内容</td>
```

其中，colspan 属性的取值为数值型整数数据，代表几个单元格进行左右合并。

例如，在上面表格的基础上将 A1 和 B1 单元格合并成一个单元格，为第一行的第一个<td>标记增加 colspan="2"属性，并且将 B1 单元格的<td>标记删除。

【例 4.2】（实例文件：ch04\4.2.html）

```
<!DOCTYPE html>
<html>
<head>
<title>单元格左右合并</title>
</head>
<body>
<table border="1">                          <!--设置表格边框的粗细-->
  <tr>
  <td colspan="2">A1 B1</td>                <!--合并第一个行的列1和列2单元格-->
  <td>C1</td>
  </tr>
```

```
    <tr>
    <td>A2</td>
    <td>B2</td>
    <td>C2</td>
    </tr>
    <tr>
    <td>A3</td>
    <td>B3</td>
    <td>C3</td>
    </tr>
    <tr>
    <td>A4</td>
    <td>B4</td>
    <td>C4</td>
    </tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 4-3 所示。

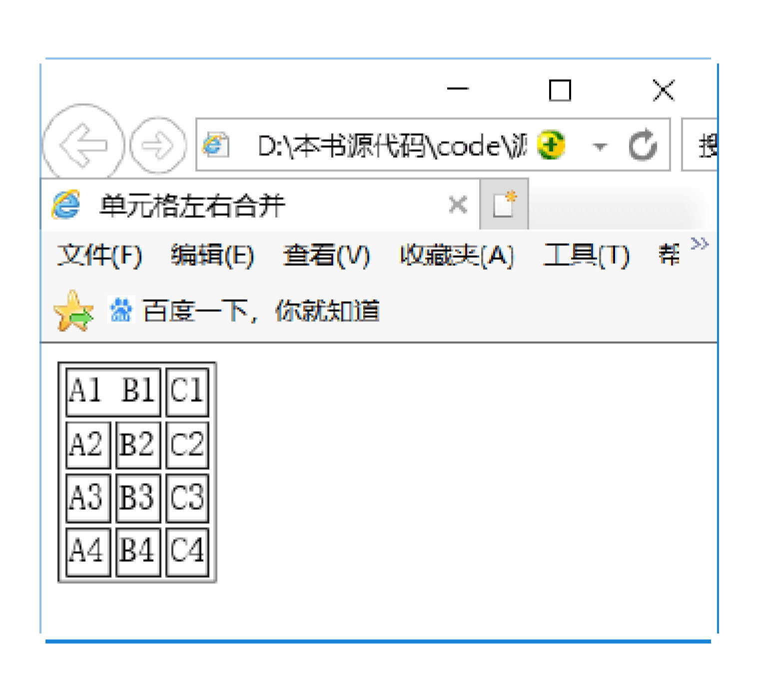


图 4-3 单元格左右合并

从预览图中可以看到，A1 和 B1 单元格合并成一个单元格，C1 还在原来的位置上。



合并单元格以后，相应的单元格标记就应该减少。例如，A1 和 B1 单元格合并后，B1 单元格的<td></td>标记应该删除，否则单元格就会多出一个，并且后面单元格依次向右位移。

2. 用 rowspan 属性合并上下单元格

上下单元格的合并需要为<td>标记增加 rowspan 属性，格式如下：

```
<td rowspan="数值">单元格内容</td>
```

其中，rowspan 属性的取值为数值型整数数据，代表几个单元格进行上下合并。

例如，在上面表格的基础上将 A1 和 A2 单元格合并成一个单元格，为第一行的第一个<td>

标记增加 rowspan="2"属性，并且将 A2 单元格的<td>标记删除。

【例 4.3】（实例文件：ch04\4.3.html）

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>单元格左右合并</title>
</head>
<body>
<table border="1">                <!--设置表格边框的粗细-->
  <tr>
  <td rowspan="2">A1A2</td>      <!--合并第1个行和第2行的列1和列2单元格-->
  <td>B1</td>
  <td>C1</td>
  </tr>
  <tr>
  <td>B2</td>
  <td>C2</td>
  </tr>
  <tr>
  <td>A3</td>
  <td>B3</td>
  <td>C3</td>
  </tr>
  <tr>
  <td>A4</td>
  <td>B4</td>
  <td>C4</td>
  </tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 4-4 所示。

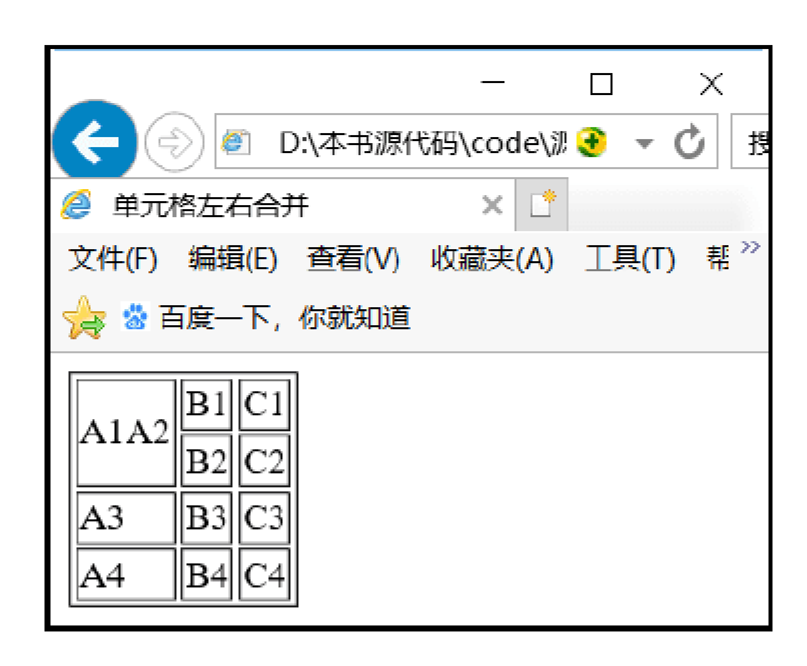


图 4-4　单元格上下合并

从预览图中可以看到，A1 和 A2 单元格合并成一个单元格。

通过上面对左右单元格合并和上下单元格合并的操作，读者会发现合并单元格的实质就是“丢掉”某些单元格。对于左右合并，就是以左侧为准，将右侧要合并的单元格“丢掉”；对于上下合并，就是以上方为准，将下方要合并的单元格“丢掉”。如果一个单元格既要向右合并又要向下合并，那么该如何实现呢？

【例 4.4】（实例文件：ch04\4.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>单元格左右合并</title>
</head>
<body>
<table border="1">                                     <!--设置表格边框的粗细-->
  <tr>
  <td colspan="2" rowspan="2">A1B1<br>A2B2</td>   <!--既要向右合并，又要向下
合并-->
  <td>C1</td>
  </tr>
  <tr>
  <td>C2</td>
  </tr>
  <tr>
  <td>A3</td>
  <td>B3</td>
  <td>C3</td>
  </tr>
  <tr>
  <td>A4</td>
  <td>B4</td>
  <td>C4</td>
  </tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览网页效果如图 4-5 所示。

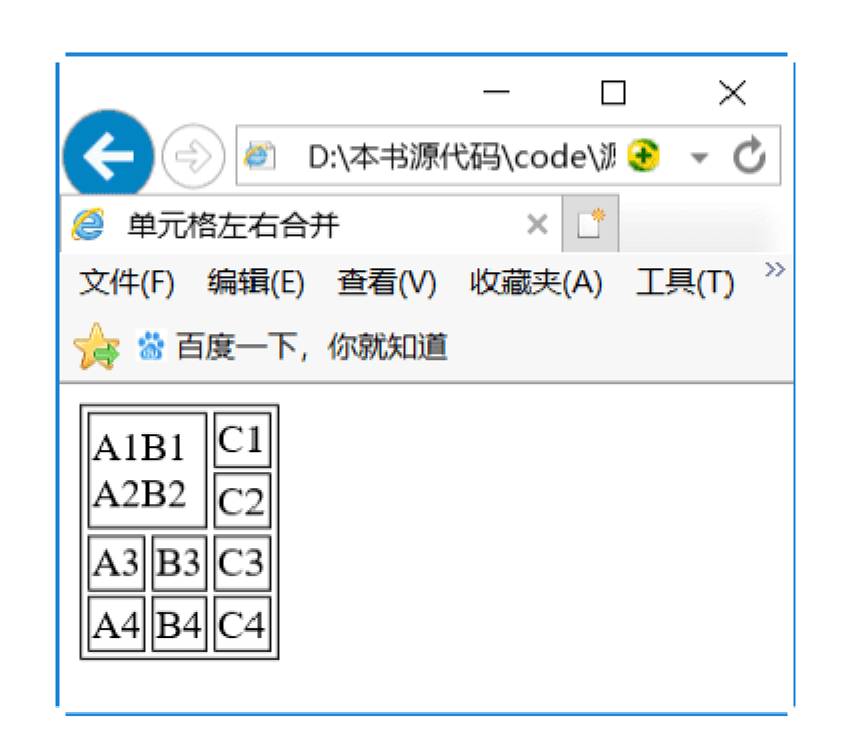


图 4-5　两个方向合并单元格

从上面的代码中可以看到，A1 单元格向右合并 B1 单元格，向下合并 A2 单元格，并且 A2 单元格向右合并 B2 单元格。

3. 使用 Dreamweaver CC 合并单元格

使用 HTML 创建表格非常麻烦，Dreamweaver CC 工具提供了表格的快捷操作，类似于在 Word 工具中编辑表格的操作。在 Dreamweaver CC 中创建表格，只需要执行【插入】菜单中的【表格】命令，在弹出的对话框中指定表格的行数、列数、宽度和边框，即可在光标处创建一个空白表格。选择表格之后，【属性】面板提供了表格的常用操作，如图 4-6 所示。

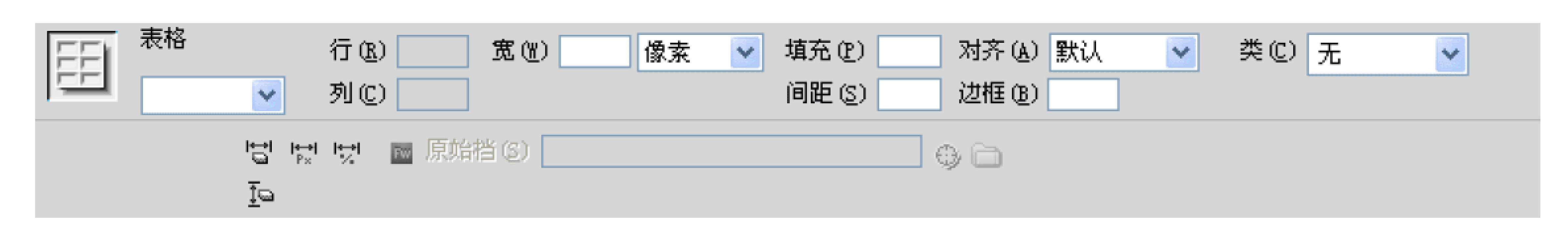


图 4-6　表格【属性】面板



表格【属性】面板中的操作，请结合前面讲述的 HTML 语言。对于按钮命令，请读者将鼠标悬停于按钮之上，数秒之后会出现命令提示。

关于表格的操作不再赘述，请读者自行操作，这里重点讲解如何使用 Dreamweaver CC 合并单元格。在 Dreamweaver CC 可视化操作中，提供了合并与拆分单元格两种操作。拆分单元格的操作，其实还是逆行的合并操作。进行单元格合并和拆分时，请将光标置于单元格内，如果选择了一个单元格，则拆分命令有效（如图 4-7 所示）；如果选择了两个或两个以上单元格，则合并命令有效。

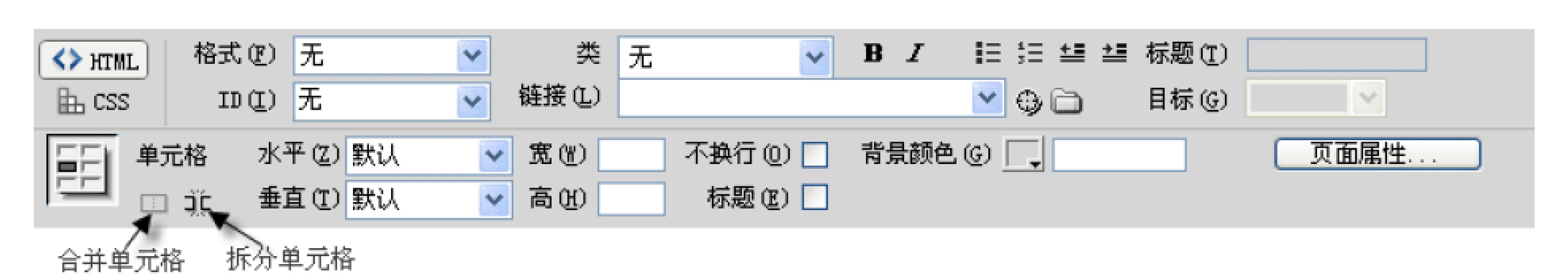


图 4-7　拆分单元格有效

4.2　完整的表格标记

前面讲述了表格中常用的 3 个标记：<table>、<tr>和<td>，使用它们可以构建出最简单的表格。为了让表格结构更清楚，以及配合后面学习的 CSS 样式，更方便地制作各种样式的表格，表格中还会出现表头、主体、脚注等。

按照表格结构，可以把表格的行分组，称为“行组”，不同的行组具有不同的意义。行组

分为三类——“表头”“主体”和“脚注”。三者相应的 HTML 标记依次为<thead>、<tbody>和<tfoot>。

此外，在表格中还有两个标记。标记<caption>表示表格的标题，在一行中除了<td>标记表示一个单元格以外，还可以使用<th>定义表格内的表头单元格。

【例 4.5】（实例文件：ch04\4.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>完整表格标记</title>
<style>
tfoot{
    background-color:#FF3;
}
</style>
</head>
<body>
<table border="1">                          <!--设置表格边框的粗细-->
  <caption>学生成绩单</caption>
  <thead>
  <tr>
    <th>姓名</th><th>性别</th><th>成绩</th>
  </tr>
  </thead>
   <tfoot>
  <tr>
    <td>平均分</td><td colspan="2">540</td>
  </tr>
  </tfoot>
  <tbody>
  <tr>
    <td>张三</td><td>男</td><td>560</td>
  </tr>
  <tr>
    <td>李四</td><td>男</td><td>520</td>
  </tr>
 </tbody>
</table>
</body>
</html>
```

从上面的代码可以发现，使用<caption>标记定义了表格标题，<thead>、<tbody>和<tfoot>标记对表格进行了分组。在<thead>部分使用<th>标记代替<td>标记定义单元格，<th>标记定义的单元格默认加粗。网页预览效果如图 4-8 所示。

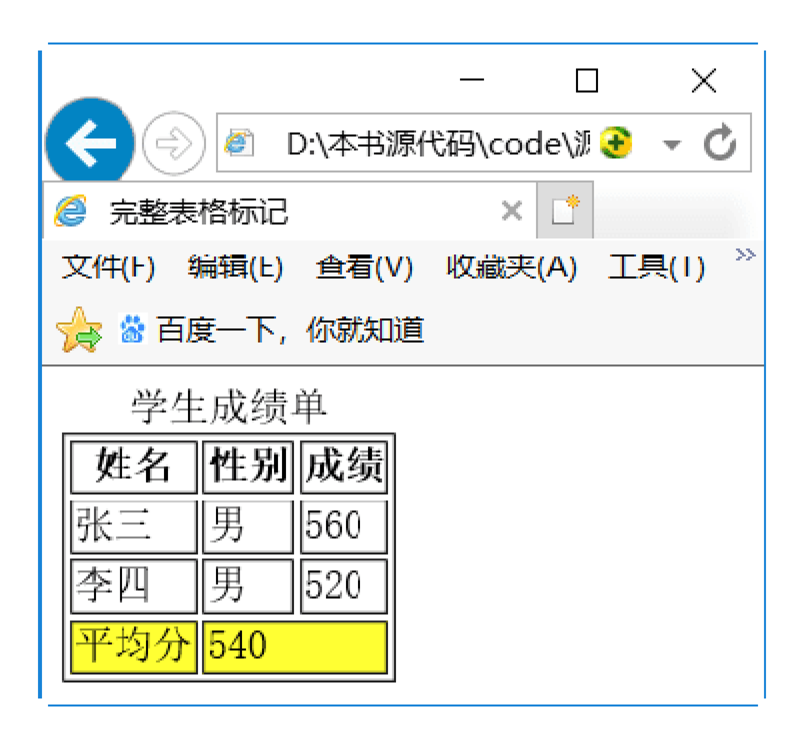


图 4-8 完整的表格结构



<caption> 标记必须紧随<table> 标记之后。

4.3 综合实例——制作计算机报价单

利用所学的表格知识，制作如图 4-9 所示的计算机报价单。



图 4-9 计算机报价单

具体操作步骤如下：

01 新建 HTML 文档并对其简化，代码如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>完整表格标记</title>
</head>
<body>
</body>
</html>
```

02 保存 HTML 文件，选择相应的保存位置，文件名为“综合实例——制作计算机报价单.html”。

03 在 HTML 文档的 body 部分增加表格及内容，代码如下：

```
<table>
   <caption>计算机报价单</caption>
  <tr>
    <th>型号</th>
    <th>类型</th>
    <th>价格</th>
    <th>图片</th>
  </tr>
  <tr>
    <td>宏碁 (Acer) AS4552-P362G32MNCC</td>
    <td>笔记本</td>
    <td>￥2799</td>
    <td><img src="images/Acer.jpg" width="120" height="120"></td>
  </tr>
  <tr>
<td>戴尔 (Dell) 14VR-188</td>
<td>笔记本</td>
    <td>￥3499</td>
    <td><img src="images/Dell.jpg" width="120" height="120"></td>
  </tr>
   <tr>
    <td>联想 (Lenovo) G470AH2310W42G500P7CW3(DB)-CN  </td>
    <td>笔记本</td>
    <td>￥4149</td>
    <td><img src="images/Lenovo.jpg" width="120" height="120"></td>
  </tr>
  <tr>
    <td>戴尔家用 (DELL)  I560SR-656</td>
    <td>台式</td>
    <td>￥3599</td>
    <td><img src="images/DellT.jpg" width="120" height="120"></td>
  </tr>
  <tr>
```

```
      <td>宏图奇眩(Hiteker)  HS-5508-TF</td>
      <td>台式</td>
      <td>￥3399</td>
      <td><img src="images/Hiteker.jpg" width="120" height="120"></td>
    </tr>
    <tr>
      <td>联想 (Lenovo) G470</td>
      <td>笔记本</td>
      <td>￥4299</td>
      <td><img src="images/LenovoG.jpg" width="120" height="120"></td>
    </tr>
  </table>
```

利用<caption>标记制作表格的标题，<th>代替<td>作为标题行单元格。可以将图片放在单元格内，即在<td>标记内使用<img>标记。

04 在 HTML 文档的 head 部分增加 CSS 样式，为表格增加边框及相应的修饰，代码如下：

```
<style>
table{
    /*表格增加线宽为3的橙色实线边框*/
    border:3px solid #F60;
}
caption{
    /*表格标题字号36*/
    font-size:36px;
}
th,td{
    /*表格单元格（th、td）增加边线*/
    border:1px solid #F90;
}
</style>
```

05 保存网页后，即可查看最终效果。

4.4　专家解惑

1. 在 Dreamweaver CC 中如何选择多个单元格？

答：在 Dreamweaver CC 中选择单元格的操作类似于文字处理工具 Word，按住鼠标左键拖动，经过的单元格都会被选择。按住 Ctrl 键的同时依次单击某个单元格，这些单元格将会被选择，选择的单元格可以是不连续的。在需要选择区域的开头单元格中单击，按住 Shift 键，在区域的末尾单元格中单击，开头和结尾单元格组成的区域内的所有单击格将会被选择。

2. 表格除了显示数据，还可以进行布局，为何不使用表格进行布局？

答：在互联网刚刚开始普及时，网页非常简单，形式也非常单调，当时美国设计师 David Siegel 发明了使用表格布局，风靡全球。在表格布局的页面中，表格不但需要显示内容，还要控制页面的外观及显示位置，导致页面代码过多，结构与内容无法分离。这样就给网站的后期维护和其他方面带来了麻烦。

3. 使用<thead>、<tbody>和<tfoot>标记对行进行分组的意义何在？

答：在 HTML 文档中增加<thead>、<tbody>和<tfoot>标记虽然从外观上不能看出任何变化，但是它们却使文档的结构更加清晰。使用<thead>、<tbody>和<tfoot>标记除了使文档更加清晰之外，还有一个更重要的意义，那就是方便使用 CSS 样式对表格的各个部分进行修饰，从而制作出更炫的表格。

第5章 使用表单

在网页中，表单主要是用于负责采集浏览者的相关数据，如常见的注册表、调查表和留言表等。在 HTML5 中，表单拥有多个新的表单输入类型，这些新特性提供了更好的输入控制和验证。本章主要讲述表单概述、表单基本元素的使用方法和表单高级元素的使用方法，最后通过一个综合案例，进一步讲述表单的实用技巧。

5.1 表单概述

表单主要用于收集网页上浏览者的相关信息，其标记为<form></form>。表单的基本语法格式如下：

```
<form action="url" method="get|post" enctype="mime"></form>
```

其中，action 指定处理提交表单的格式，它可以是一个 URL 地址或一个电子邮件地址；method 指明提交表单的 HTTP 方法。enctype 指明用来把表单提交给服务器时的互联网媒体形式。

表单是一个能够包含表单元素的区域，通过添加不同的表单元素，将显示不同的效果。

【例 5.1】（实例文件：ch05\5.1.html）

```
<!DOCTYPE html>
<html>
<body>
<form>
下面是输入用户登录信息<br />
用户名称
<input type="text" name="user"><br />
用户密码
<input type="password" name="password"><br />
<input type="submit" value="登录">
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-1 所示，可以看到用户登录信息页面。

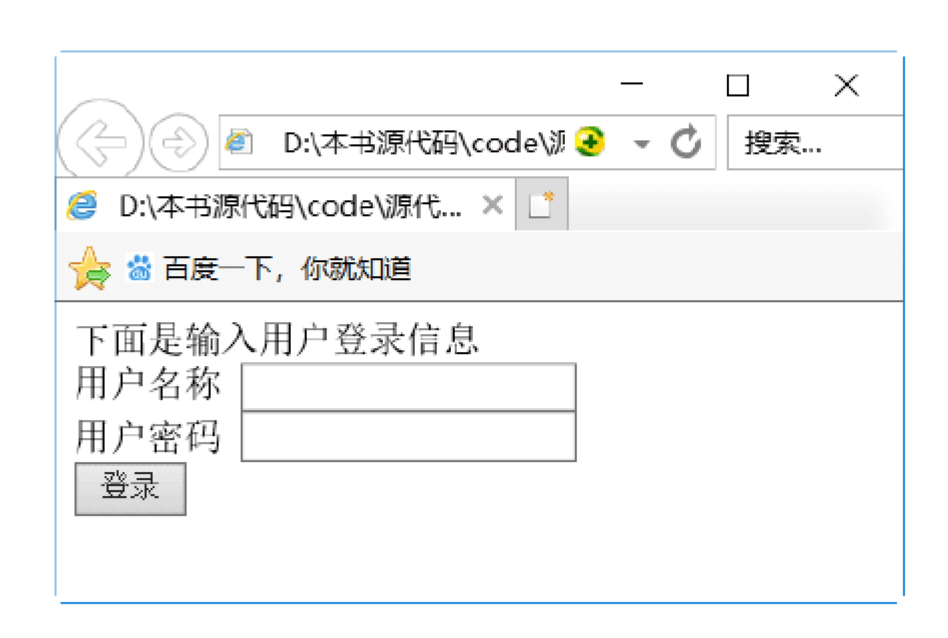


图 5-1　用户登录页面

5.2　表单基本元素的使用

表单元素是能够让用户在表单中输入信息的元素，常见的有文本框、密码框、下拉菜单、单选按钮、复选框等。

5.2.1　单行文本输入框 text

文本框是一种让访问者自行输入内容的表单对象，通常被用来填写单个字或简短的回答，如用户姓名和地址。代码格式如下：

```
<input type="text" name="..." size="..." maxlength="..." value="...">
```

其中，type="text"定义单行文本输入框；name 属性定义文本框的名称，要保证数据的准确采集，必须定义一个独一无二的名称；size 属性定义文本框的宽度，单位是单个字符宽度；maxlength 属性定义最多输入的字符数；value 属性定义文本框的初始值。

【例 5.2】（实例文件：ch05\5.2.html）

```
<!DOCTYPE html>
<html>
<head><title>输入用户的姓名</title></head>
<body>
<form>
请输入您的姓名：
<input type="text" name="yourname" size="20" maxlength="15"><br />
请输入您的地址：
<input type="text" name="youradr" size="20" maxlength="15">
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-2 所示，可以看到两个单行文本输入框。

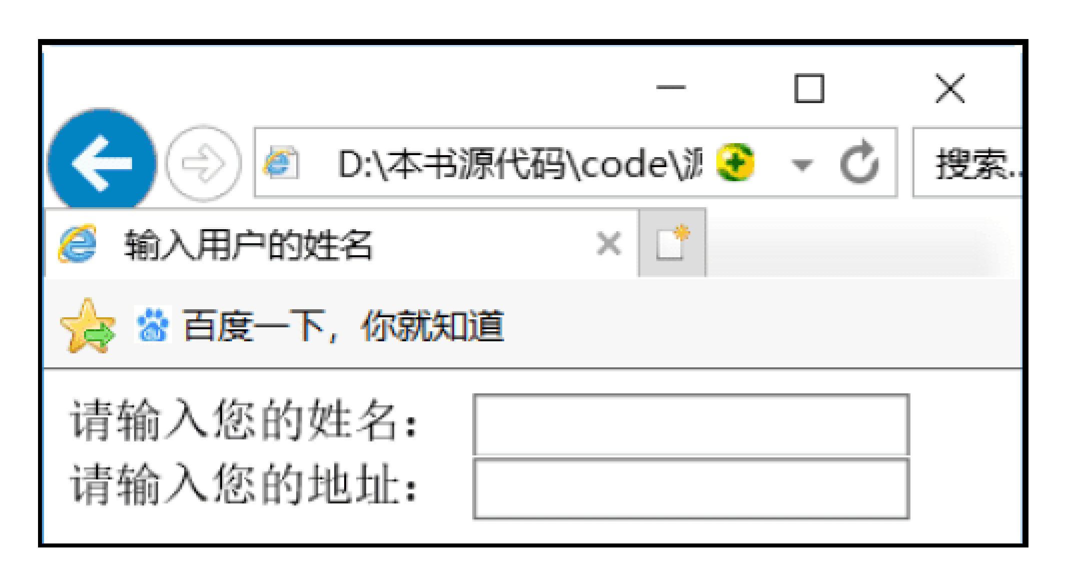


图 5-2 单行文本输入框

5.2.2 多行文本框标记<textarea>

多行文本框标记<textarea>主要用于输入较长的文本信息。代码格式如下：

```
<textarea name="..." cols="..." rows="..." wrap="..."></textarea >
```

其中，name 属性定义多行文本框的名称，要保证数据的准确采集，必须定义一个独一无二的名称；cols 属性定义多行文本框的宽度，单位是单个字符宽度；rows 属性定义多行文本框的高度，单位是单个字符高度；wrap 属性定义输入内容大于文本域时显示的方式。

【例 5.3】（实例文件：ch05\5.3.html）

```
<!DOCTYPE html>
<html>
<head><title>多行文本输入</title></head>
<body>
<form>
请输入您最新的工作情况<br />
<textarea name="yourworks" cols="50" rows="5"></textarea>
<br />
<input type="submit" value="提交">
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-3 所示，可以看到多行文本框。

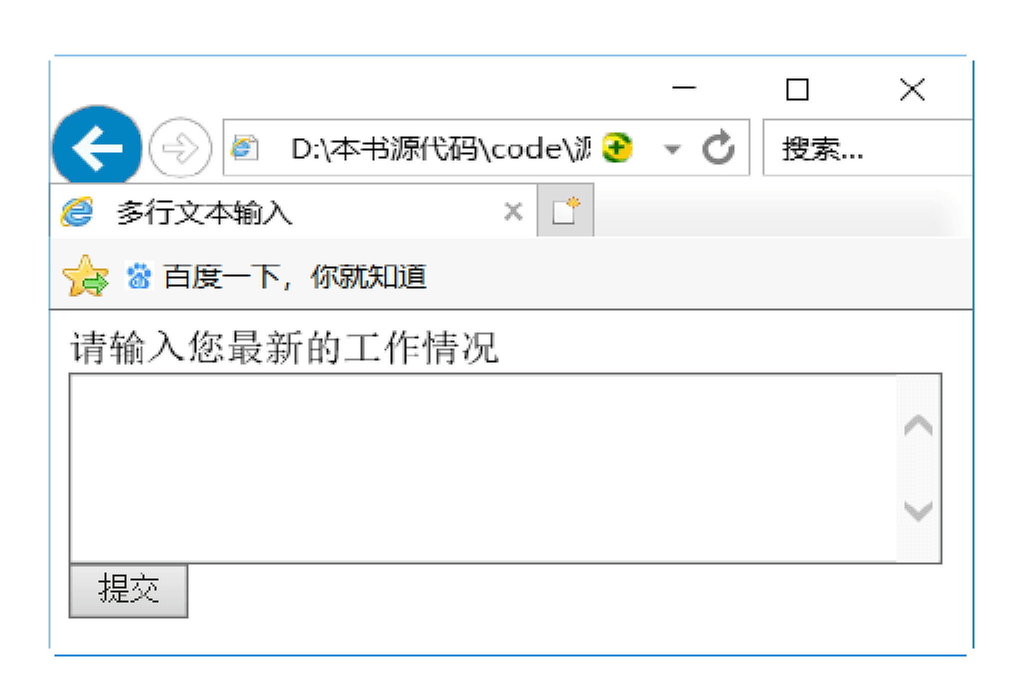


图 5-3 多行文本框

5.2.3 密码域 password

密码输入框是一种特殊的文本域，主要用于输入一些保密信息。当网页浏览者输入文本时，显示的是黑点或其他符号，这样就增加了输入文本的安全性。代码格式如下：

```
<input type="password" name="..." size="..." maxlength="...">
```

其中，type="password"定义密码框；name 属性定义密码框的名称，要保证唯一性；size 属性定义密码框的宽度，单位是单个字符宽度；maxlength 属性定义最多输入的字符数。

【例 5.4】（实例文件：ch05\5.4.html）

```
<!DOCTYPE html>
<html>
<head><title>输入用户姓名和密码 </title></head>
<body>
<form >
用户姓名：
<input type="text" name="yourname">
<br />
登录密码：
<input type="password" name="yourpw"><br />
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-4 所示，输入用户名和密码时可以看到密码以黑点的形式显示。

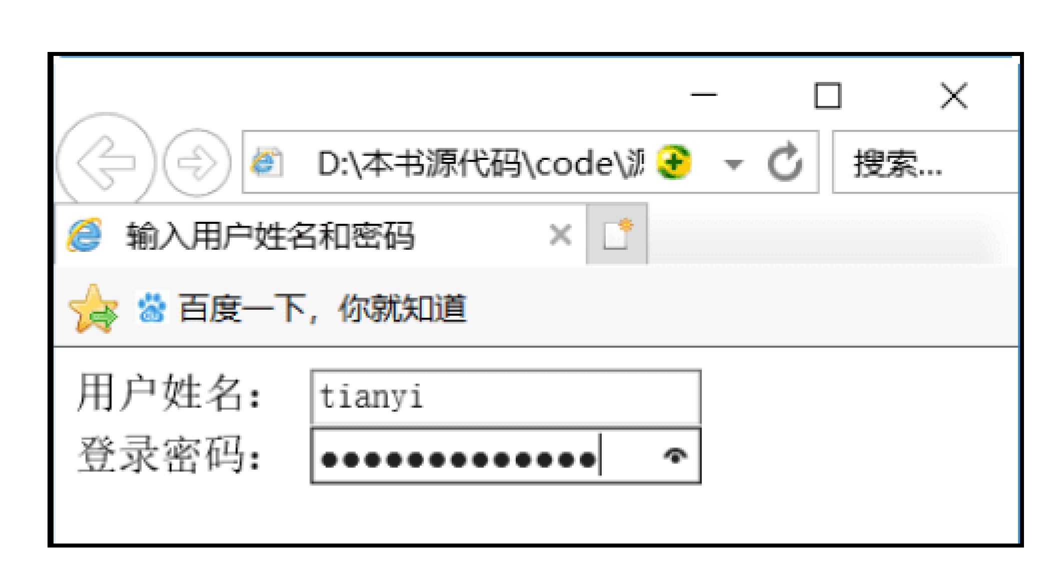


图 5-4 密码输入框

5.2.4 单选按钮 radio

单选按钮主要是让网页浏览者在一组选项里只能选其一。代码格式如下：

```
<input type="radio" name="..." value = "...">
```

其中，type="radio"定义单选按钮；name 属性定义单选按钮的名称，单选按钮都是以组为单位使用的，在同一组中的单选项都必须用同一个名称；value 属性定义单选按钮的值，在同

一组中它们的域值必须是不同的。

【例 5.5】（实例文件：ch05\5.5.html）

```
<!DOCTYPE html>
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form >
请选择您感兴趣的图书类型：
<br />
<input type="radio" name="book" value="Book1">网站编程<br />
<input type="radio" name="book" value="Book2">办公软件<br />
<input type="radio" name="book" value="Book3">设计软件<br />
<input type="radio" name="book" value="Book4">网络管理<br />
<input type="radio" name="book" value="Book5">黑客攻防<br />
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-5 所示，即可看到 5 个单选按钮，用户只能选中一个单选按钮。

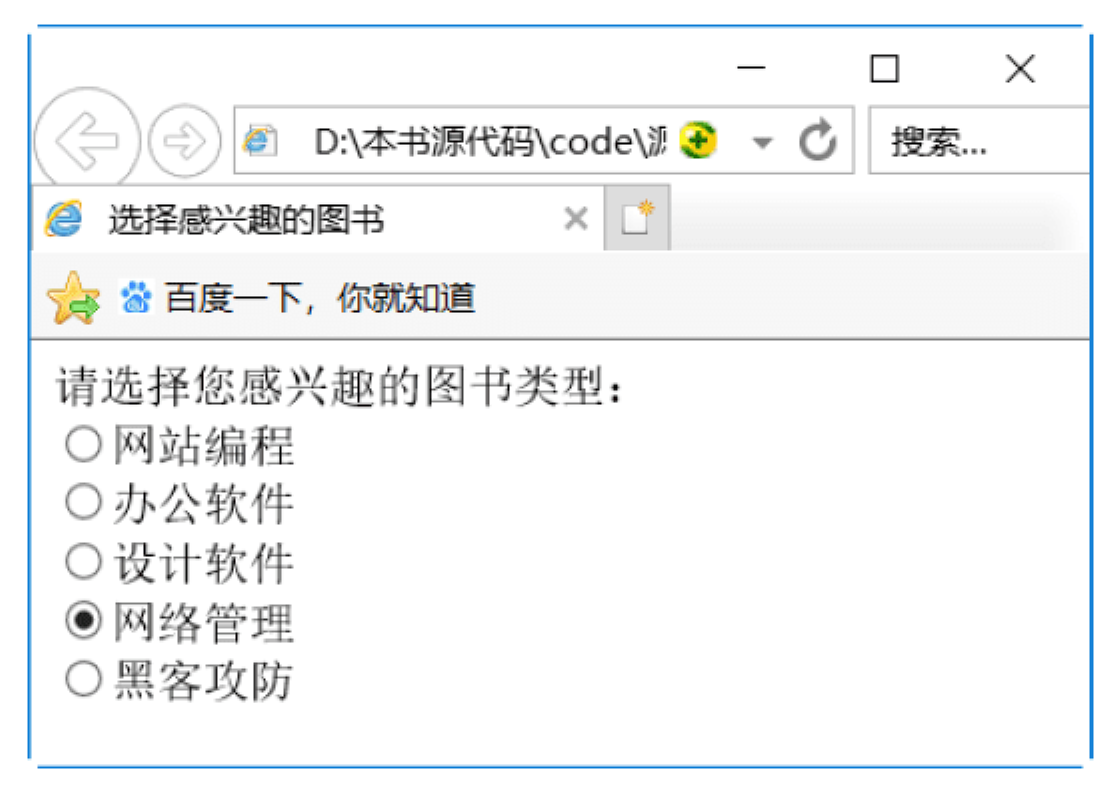


图 5-5　单选按钮

5.2.5　复选框 checkbox

复选框主要是让网页浏览者在一组选项里可以同时选择多个选项。每个复选框都是一个独立的元素，必须有一个唯一的名称。代码格式如下：

```
<input type="checkbox" name="…" value ="…">
```

其中，type="checkbox"定义复选框；name 属性定义复选框的名称，在同一组中的复选框都必须用同一个名称；value 属性定义复选框的值。

【例 5.6】（实例文件：ch05\5.6.html）

```
<!DOCTYPE html>
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form >
请选择您感兴趣的图书类型：<br />
<input type="checkbox" name="book" value="Book1">网站编程<br />
<input type="checkbox" name="book" value="Book2">办公软件<br />
<input type="checkbox" name="book" value="Book3">设计软件<br />
<input type="checkbox" name="book" value="Book4">网络管理<br />
<input type="checkbox" name="book" value="Book5" checked>黑客攻防<br />
</form>
</body>
</html>
```



checked 属性主要用来设置默认选中项。

在 IE 11.0 中浏览效果如图 5-6 所示，即可看到 5 个复选框，其中【黑客攻防】复选框被默认选中。

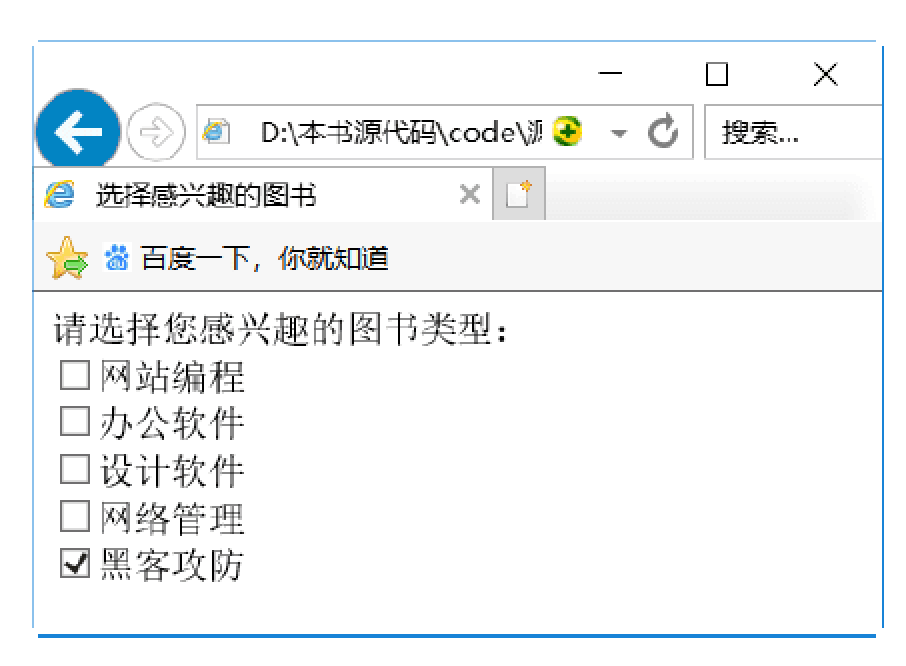


图 5-6　复选框

5.2.6　选择列表标记<select>

下拉列表主要用于在有限的空间里设置多个选项，它既可以用作单选，也可以用作多选。代码格式如下：

```
<select name="..." size="..." multiple>
<option value="..." selected>
...
</option>
 ...
</select>
```

其中，name 属性定义选择列表的名称；size 属性定义选择列表的行数；multiple 属性表示可以多选，如果不设置该属性，则只能单选；value 属性定义选择项的值；selected 属性表示默认已经选择本选项。

【例 5.7】（实例文件：ch05\5.7.html）

```
<!DOCTYPE html>
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form>
请选择您感兴趣的图书类型：<br />
<select name="fruit" size="3" multiple>
<option value="Book1">网站编程
<option value="Book2">办公软件
<option value="Book3">设计软件
<option value="Book4">网络管理
<option value="Book5">黑客攻防
</select>
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-7 所示，列表内显示了 3 个选项，用户可以按住 Ctrl 键选择多个选项。

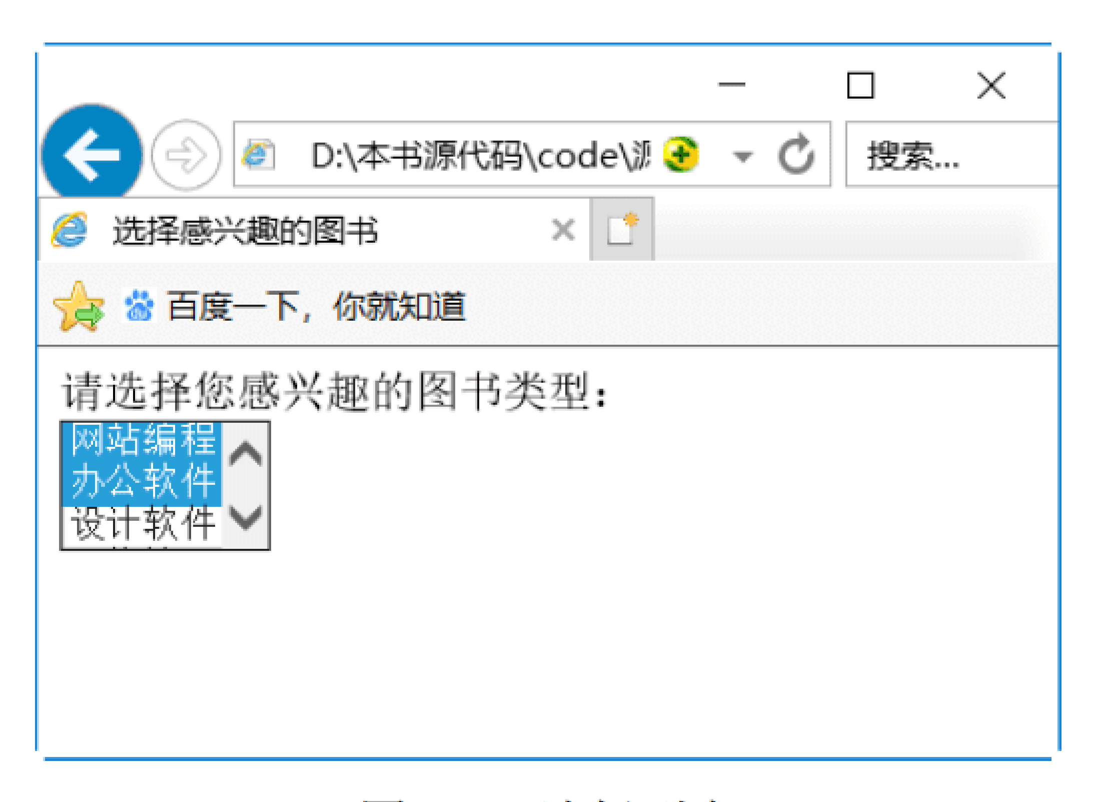


图 5-7 选择列表

5.2.7 普通按钮 button

普通按钮用来控制其他定义了脚本的处理工作。代码格式如下：

```
<input type="button" name="..." value="..." onclick="...">
```

其中，type="button"定义普通按钮；name 属性定义普通按钮的名称；value 属性定义按钮

的显示文字；onclick 属性表示单击行为，也可以通过指定脚本函数来定义按钮的行为。

【例 5.8】（实例文件：ch05\5.8.html）

```
    <!DOCTYPE html>
    <html>
    <body>
    <form>
    点击下面的按钮，把文本框1的内容复制到文本框2中：
    <br/>
    文本框1: <input type="text" id="field1" value="学习HTML5的技巧">
    <br/>
    文本框2: <input type="text" id="field2">
    <br/>
    <input type="button" name="..." value="单击我"
onclick="document.getElementById('field2').value=document.getElementById('f
ield1').value">
    </form>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 5-8 所示，单击【单击我】按钮即可将文本框 1 中的内容复制到文本框 2 中。

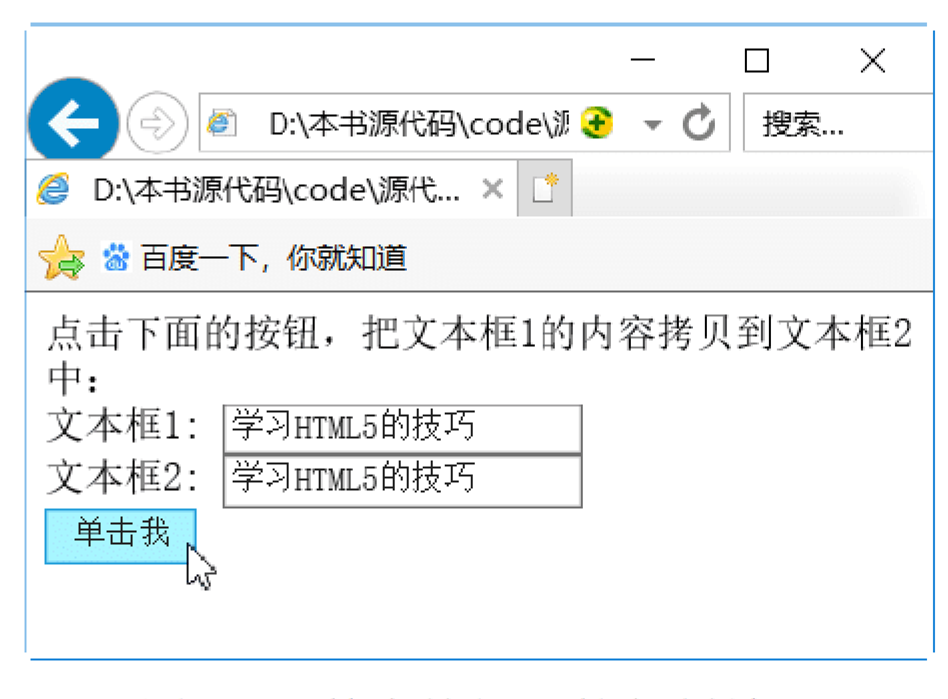


图 5-8　单击按钮后的复制效果

5.2.8　提交按钮 submit

提交按钮用来将输入的信息提交到服务器。代码格式如下：

```
    <input type="submit" name="..." value="...">
```

其中，type="submit"定义提交按钮；name 属性定义提交按钮的名称；value 属性定义按钮的显示文字。通过提交按钮可以将表单里的信息提交给 action 所指向的文件。

【例 5.9】（实例文件：ch05\5.9.html）

```
<!DOCTYPE html>
<html>
<head><title>输入用户名信息</title></head>
<body>
<form  action="http://www.yinhangit.com/yonghu.asp" method="get">
请输入您的姓名:
<input type="text" name="yourname">
<br />
请输入您的住址:
<input type="text" name="youradr">
<br />
请输入您的单位:
<input type="text" name="yourcom">
<br />
请输入您的联系方式:
<input type="text" name="yourcon">
<br />
<input type="submit" value="提交">
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-9 所示，输入内容后单击【提交】按钮，即可将表单中的数据提交到指定的服务器中。

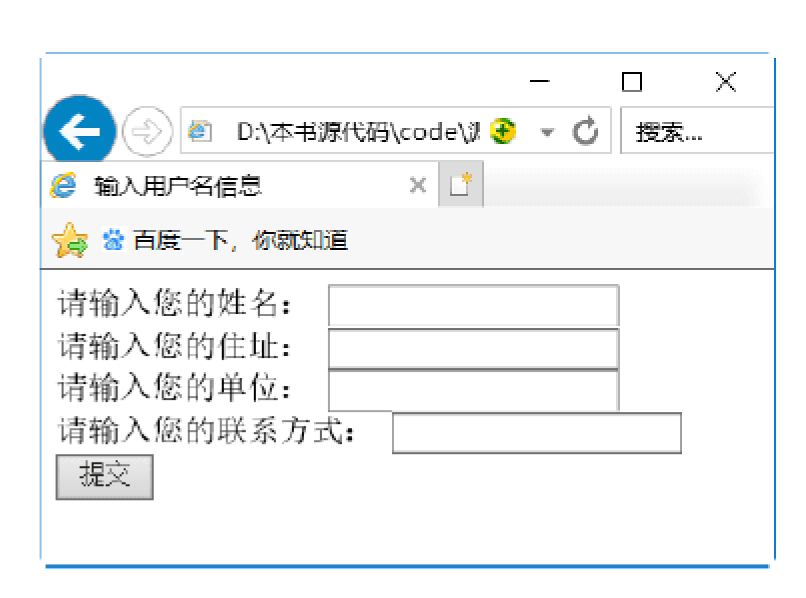


图 5-9　提交按钮

5.2.9　重置按钮 reset

重置按钮用来清空表单中输入的信息。代码格式如下：

```
<input type="reset" name="..." value="...">
```

其中，type="reset"定义重置按钮；name 属性定义重置按钮的名称；value 属性定义按钮的显示文字。

【例 5.10】（实例文件：ch05\5.10.html）

```
<!DOCTYPE html>
<html>
<body>
<form>
请输入用户名称：
<input type="text">
<br/>
请输入用户密码：
<input type="password">
<br />
<input type="submit" value="登录">
<input type="reset" value="重置">
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-10 所示，输入内容后单击【重置】按钮，即可实现将表单中的数据清空的目的。

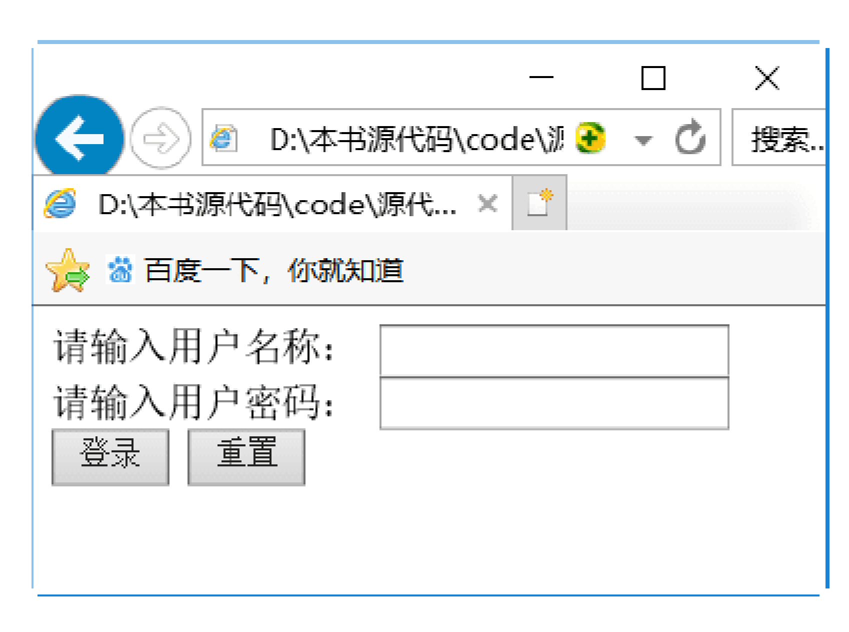


图 5-10 【重置】按钮

5.3 表单高级元素的使用

除了上述基本属性外，HTML5 中还有一些高级属性，包括 url、eamil、time、range、search 等。对于部分高级属性，IE 11.0 浏览器暂时还不支持，下面将用 Opera 11.60 浏览器查看效果。

5.3.1 url 属性

url 属性用于说明网站网址，显示为在一个文本框中输入 URL 地址，在提交表单时会自动验证 url 的值。代码格式如下：

```
<input type="url" name="userurl"/>
```

另外，用户可以使用普通属性设置 url 输入框，例如可以使用 max 属性设置其最大值、min

属性设置其最小值、step 属性设置合法的数字间隔、value 属性规定其默认值。对于另外的高级属性中同样的设置不再重复讲述。

【例 5.11】（实例文件：ch05\5.11.html）

```
<!DOCTYPE html>
<html>
<body>
<form>
<br/>
请输入网址：
<input type="url" name="userurl"/>
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-11 所示，用户即可在文本框中输入相应的网址。如果输入的 url 格式不准确，按 Enter 键后就会弹出提示信息。

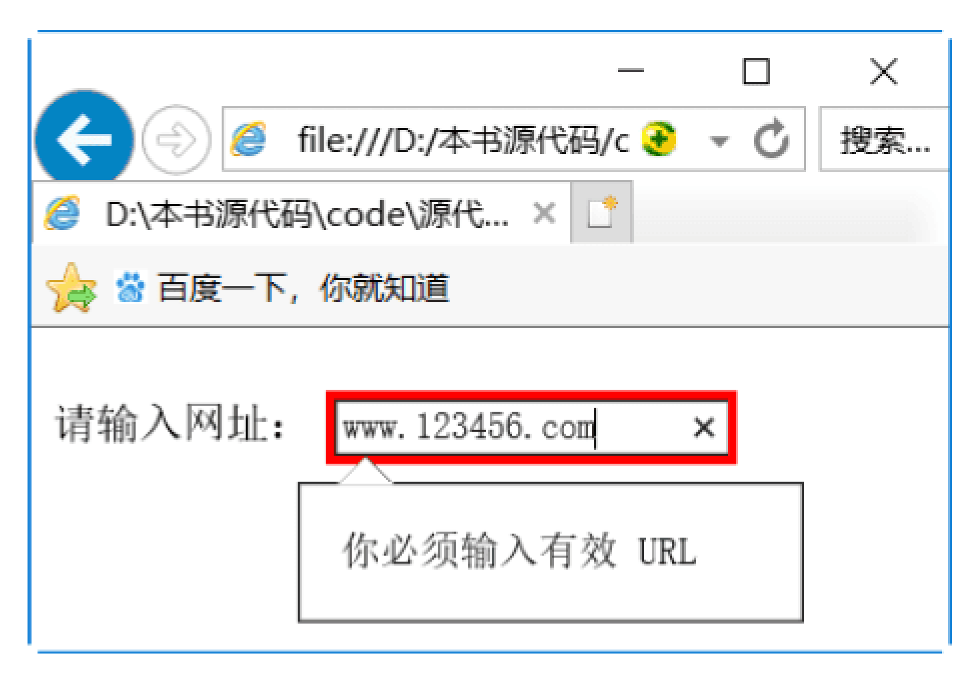


图 5-11　url 属性的效果

5.3.2　eamil 属性

与 url 属性类似，email 属性用于让浏览者输入 E-mail 地址，在提交表单时会自动验证 email 域的值。代码格式如下：

```
<input type="email" name="user_email"/>
```

【例 5.12】（实例文件：ch05\5.12.html）

```
<!DOCTYPE html>
<html>
<body>
<form>
<br/>
请输入您的邮箱地址：
<input type="email" name="user_email"/>
```

```
<br />
<input type="submit" value="提交">
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-12 所示，用户即可在文本框中输入相应的邮箱地址。如果用户输入的邮箱地址不合法，单击【提交】按钮后会弹出图中的提示信息。

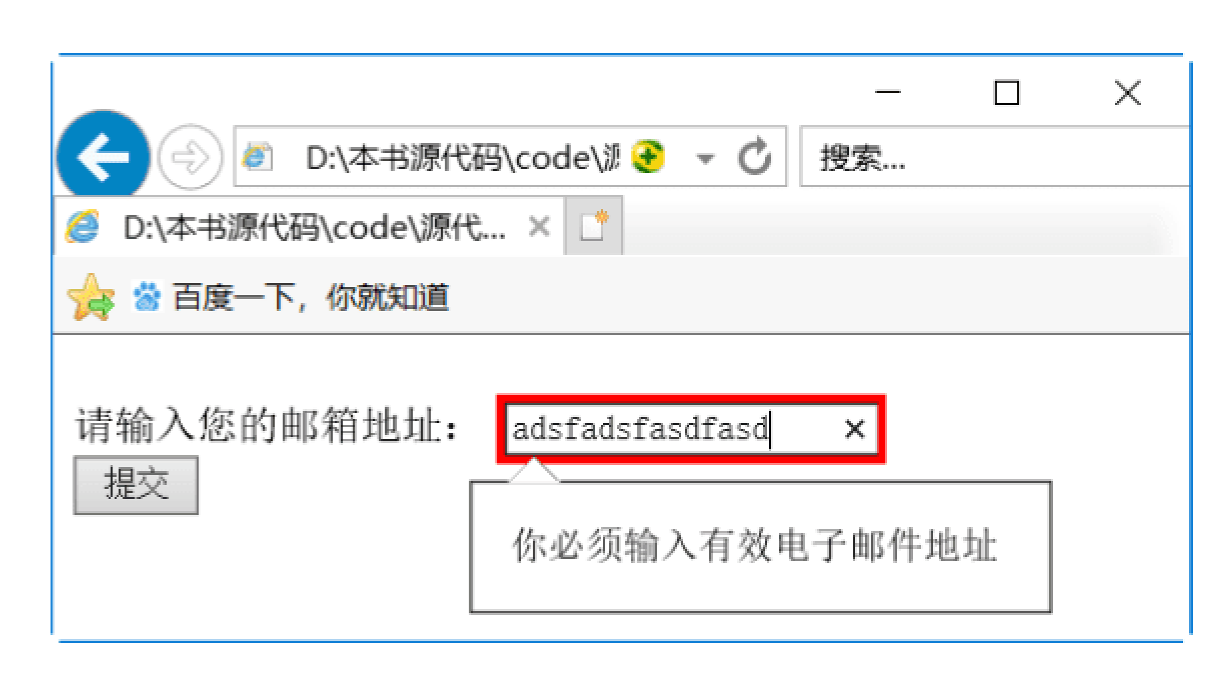


图 5-12　eamil 属性的效果

5.3.3　date 和 Times

HTML5 新增了一些日期和时间输入类型，其中包括 date、datetime、datetime-local、month、week 和 time，它们的具体含义如表 5-1 所示。

表 5-1　日期和时间输入类型

属性	含义
date	选取日、月、年
month	选取月、年
week	选取周和年
time	选取时间
datetime	选取时间、日、月、年
datetime-local	选取时间、日、月、年（本地时间）

上述属性的代码格式类似，以 date 属性为例，代码格式如下：

```
<input type="date" name="user_date" />
```

【例 5.13】（实例文件：ch05\5.13.html）

```
<!DOCTYPE html>
<html>
<body>
```

```
<form>
<br/>
请选择购买商品的日期：
<br />
<input type="date" name="user date"/>
</form>
</body>
</html>
```

在 Opera 11.60 中浏览效果如图 5-13 所示，用户单击输入框中的下三角按钮，即可在弹出的窗口中选择需要的日期。

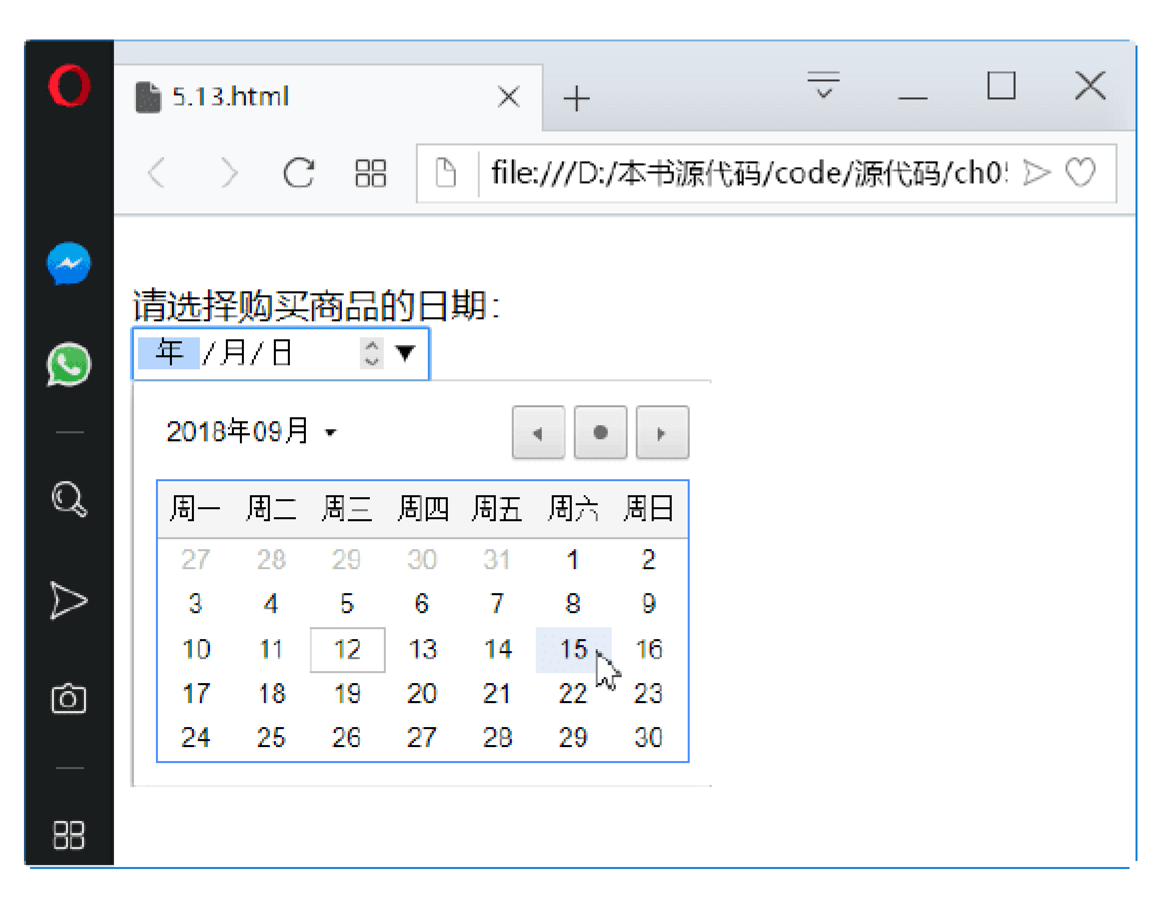


图 5-13　date 属性的效果

5.3.4　number 属性

number 属性提供了一个输入数字的输入类型，用户可以直接输入数字或通过单击微调框中的按钮选择数字。代码格式如下：

```
<input type="number" name="shuzi" />
```

【例 5.14】（实例文件：ch05\5.14.html）

```
<!DOCTYPE html>
<html>
<body>
<form>
<br/>
此网站我曾经来
<input type="number" name="shuzi"/>次了哦！
</form>
</body>
</html>
```

在 Opera 11.60 中浏览效果如图 5-14 所示，用户可以直接输入数字，也可以单击微调按钮选择合适的数字。

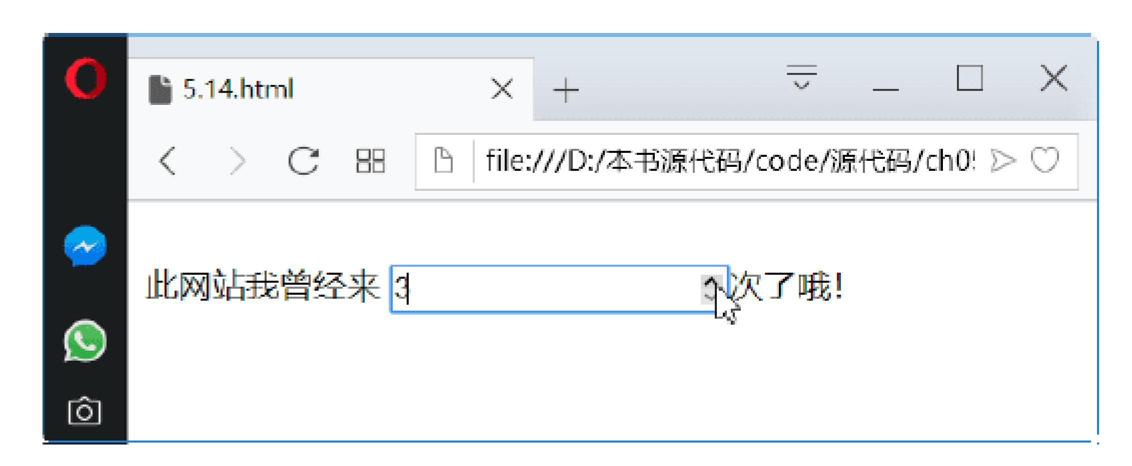


图 5-14　number 属性的效果



强烈建议用户使用 min 和 max 属性规定输入的最小值和最大值。

5.3.5　range 属性

range 属性可以显示一个滚动的控件，用户可以使用 max、min 和 step 属性设置控件的范围。代码格式如下：

```
<input type="range" name="" min="" max="" />
```

其中 min 和 max 属性分别控制滚动控件的最小值和最大值。

【例 5.15】（实例文件：ch05\5.15.html）

```
<!DOCTYPE html>
<html>
<body>
<form>
<br/>
英语成绩公布了！我的成绩名次为:
<input type="range" name="ran" min="1" max="10"/>
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-15 所示，用户可以拖曳滑块选择合适的数字。

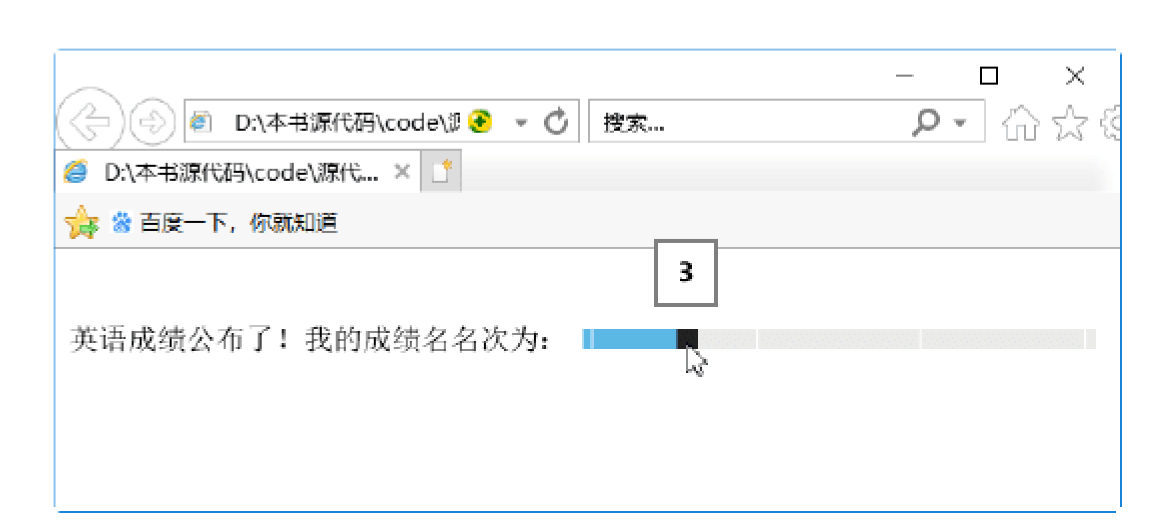


图 5-15　range 属性的效果



默认情况下，滑块位于滚珠的中间位置。如果用户指定的最大值小于最小值，则允许使用反向滚动轴，目前浏览器对这一属性还不能很好地支持。

5.3.6 required 属性

required 属性规定必须在提交之前填写输入域（不能为空）。required 属性适用于以下类型的输入属性：text、search、url、email、password、date、pickers、number、checkbox、radio 等。

【例 5.16】（实例文件：ch05\5.16.html）

```
<!DOCTYPE html>
<html>
<body>
<form>
下面是输入用户登录信息
<br />
用户名称
<input type="text" name="user" required="required">
<br />
用户密码
<input type="password" name="password" required="required">
<br />
<input type="submit" value="登录">
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-16 所示，如果用户只输入密码就单击【登录】按钮，则会弹出图中的提示信息。

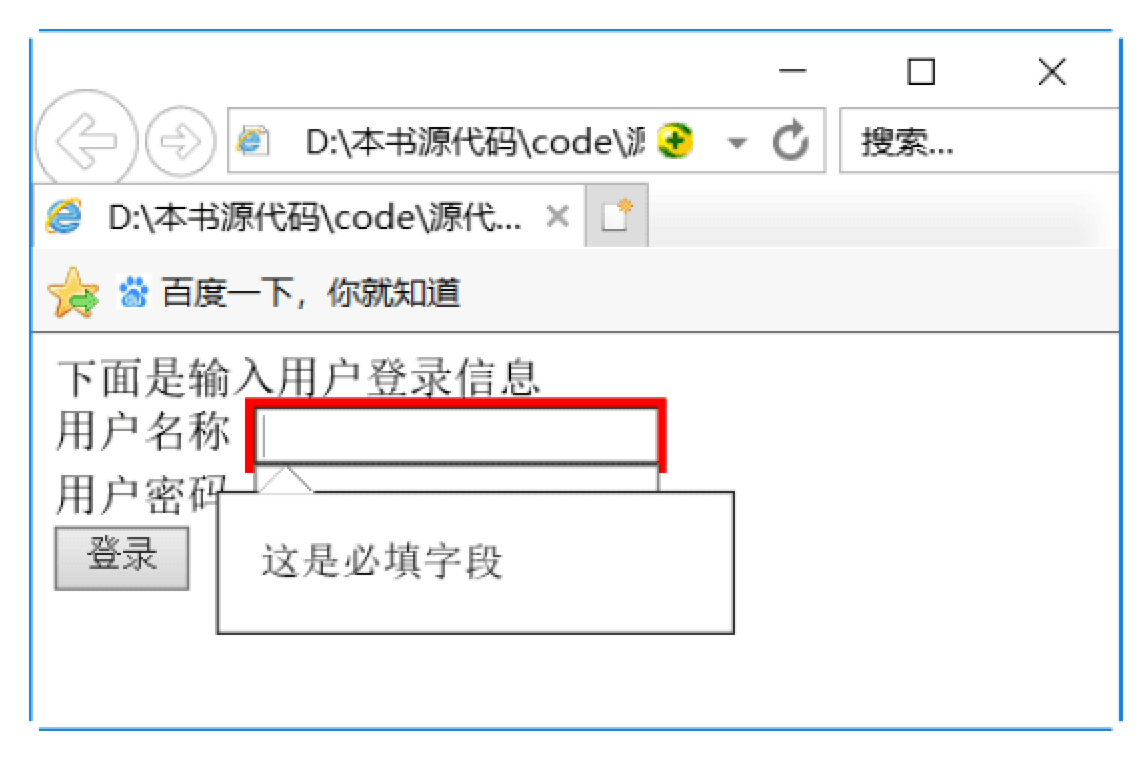


图 5-16　required 属性的效果

5.4　综合实例——创建用户反馈表单

本实例将结合使用一个表单内的各种元素，来开发一个简单网站的用户意见反馈页面。具体操作步骤如下：

01 分析需求。反馈表单非常简单，通常包含三个部分：在页面上方给出标题；标题下方是正文部分（表单元素）；最下方是表单元素提交按钮。在设计页面时，需要把“用户注册”标题设置成 h1 大小，正文使用<p>标记来限制表单元素。

02 构建 HTML 页面，实现表单内容。

```
<!DOCTYPE html>
<html>
<head>
<title>用户反馈页面</title>
</head>
<body>
<h1 align=center>用户反馈表单</h1>
<form method="post">
<p>姓    名:
<input type="text" class=txt size="12" maxlength="20" name="username"/>
</p><p>性    别:
<input type="radio" value="male"/>男
<input type="radio" value="female"/>女
</p><p>年    龄:
<input type="text" class=txt name="age"/>
</p>
<p>联系电话:
<input type="text" class=txt name="tel"/>
</p><p>电子邮件:
<input type="text" class=txt name="email"/>
</p><p>联系地址:
<input type="text" class=txt name="address"/>
</p>
<p>
请输入您对网站的建议<br />
<textarea name="yourworks" cols="50" rows="5"></textarea>
<br />
<input type="submit" name="submit" value="提交"/>
<input type="reset" name="reset" value="清除"/>
</p>
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 5-17 所示。

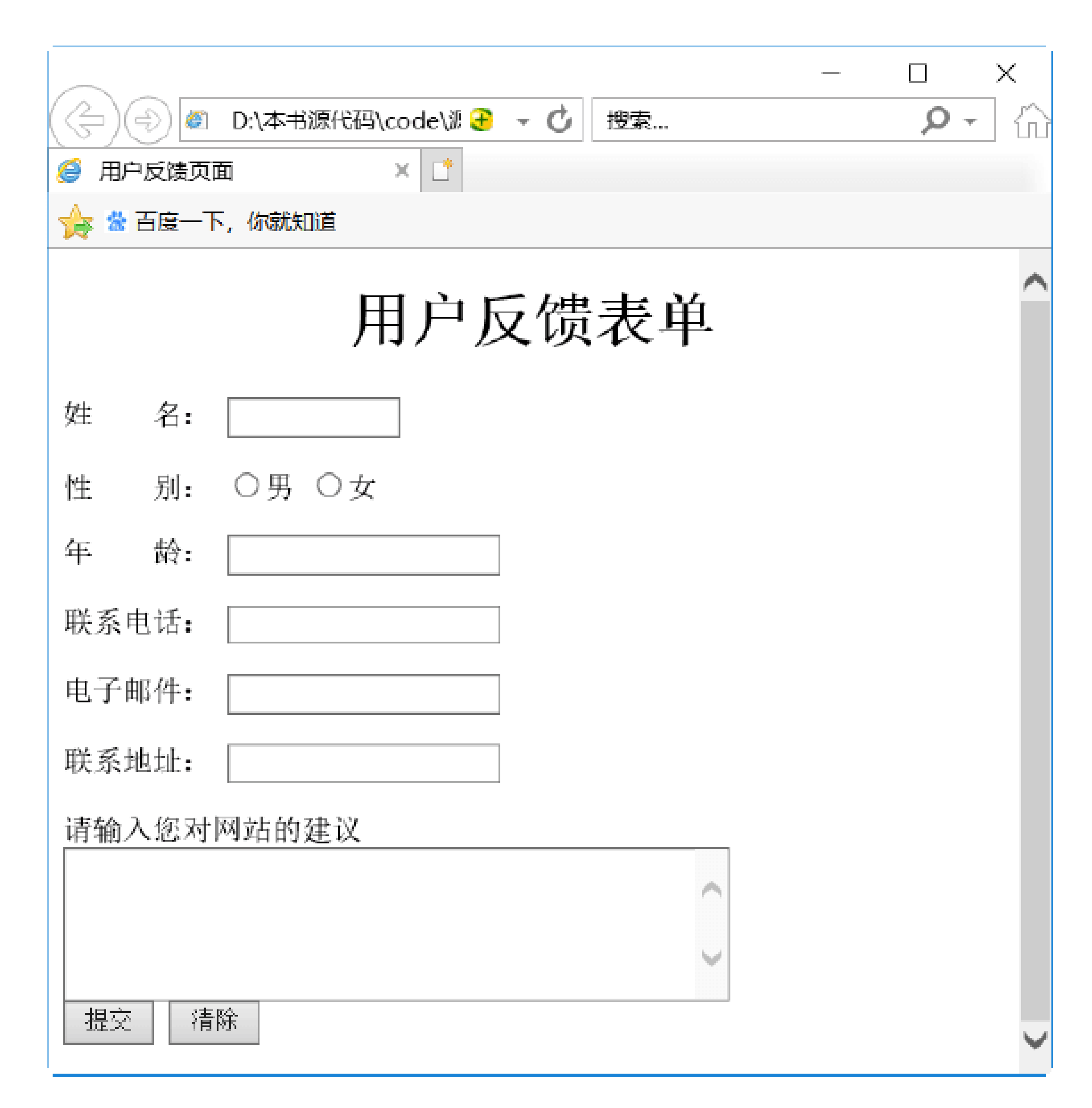


图 5-17　用户反馈页面

5.5　专家解惑

1. 如何在表单中实现文件上传框？

在 HTML5 语言中，使用 file 属性实现文件上传框。语法格式为：<input type="file" name="..." size="…" maxlength="…">。其中，type="file"定义为文件上传框；name 属性为文件上传框的名称；size 属性定义文件上传框的宽度，单位是单个字符宽度；maxlength 属性定义最多输入的字符数。文件上传框的显示效果如图 5-18 所示。

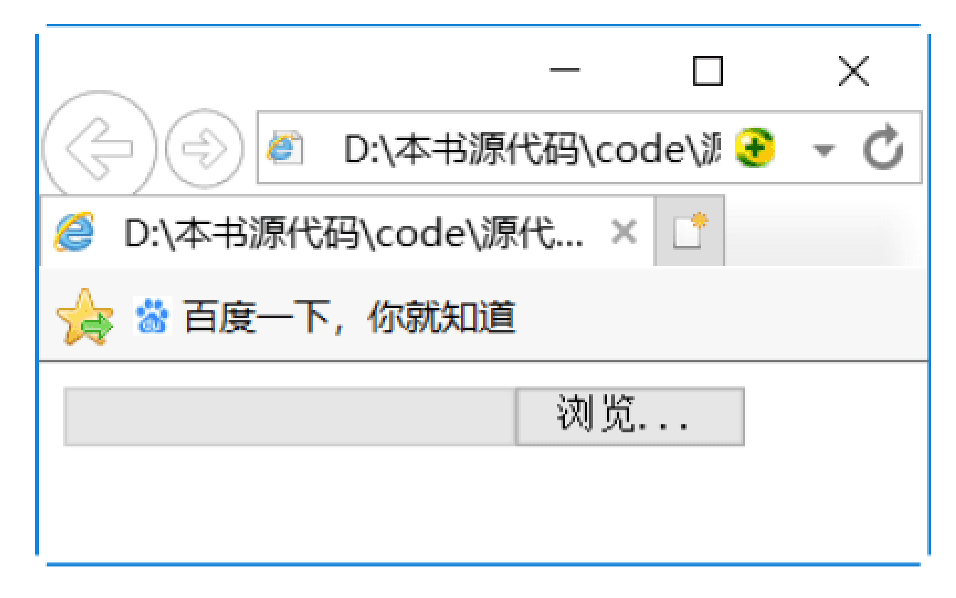


图 5-18　文件上传框

2. 制作的单选按钮为什么只能选中一个？

此时用户需要检查单选按钮的名称，保证同一组中的单选按钮名称必须相同，这样才能保证单选按钮只能选中其中一个。

第6章　HTML5中的音频和视频

目前，在网页上没有关于音频和视频的标准，多数音频和视频都是通过插件来播放的，为此，HTML5 新增了音频和视频的标记。本章将讲述音频和视频的基本概念、常用属性、解码器和浏览器的支持情况。

6.1　<audio>标记

目前，大多数音频是通过插件来播放的，如常见的播放插件 Flash，这就是为什么用户在用浏览器播放音乐时，常常需要安装 Flash 插件的原因。但是并不是所有的浏览器都拥有同样的插件。

与 HTML4 相比，HTML5 新增了<audio>标记，规定了一种包含音频的标准方法。

6.1.1　<audio>标记概述

<Audio>标记主要是定义播放声音文件或音频流的标准。支持 3 种音频格式，分别为 ogg、mp3 和 wav。

如果需要在 HTML5 网页中播放音频，则输入的基本格式如下：

```
<audio src="song.mp3" controls="controls">
</audio>
```



其中 src 属性是规定要播放的音频地址，controls 属性是提供添加播放、暂停和音量的控件。

另外，在<audio>与</audio>之间插入的内容是供不支持 audio 元素的浏览器显示的。

【例 6.1】（实例文件：ch06\6.1.html）

```
<!DOCTYPE html>
<html>
<head>
<title>audio</title>
<head>
```

```
<body>
  <audio src="song.mp3" controls="controls">
您的浏览器不支持audio标记!
</audio>
</body>
</html>
```

如果用户的浏览器是 IE 8.0 或以前的版本，浏览效果如图 6-1 所示，可见目前 IE 浏览器还不支持<audio>标记。

在 IE 11.0 中浏览效果如图 6-2 所示，可以看到加载的音频控制条，并能听到加载的音频文件。

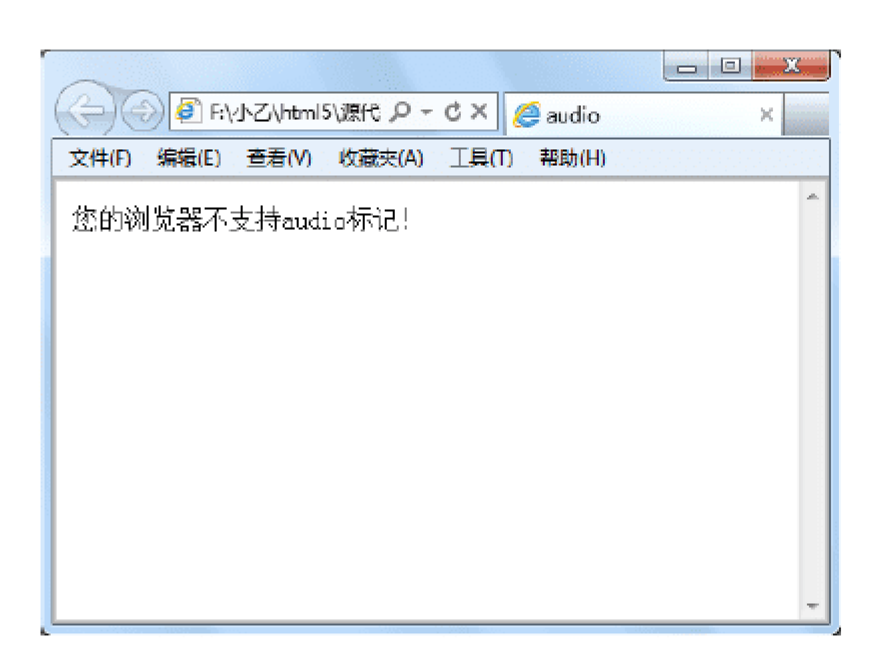


图 6-1　不支持<audio>标记的效果



图 6-2　支持<audio>标记的效果

6.1.2　<audio>标记的属性

<audio>标记的常见属性和含义如表 6-1 所示。

表 6-1　<audio>标记常见属性

属性	值	描述
autoplay	Autoplay（自动播放）	如果出现该属性，则音频在就绪后马上播放
controls	Controls（控制）	如果出现该属性，则向用户显示控件，如播放按钮
loop	Loop（循环）	如果出现该属性，则每当音频结束时重新开始播放
preload	Preload（加载）	如果出现该属性，则音频在页面加载时进行加载，并预备播放。如果使用 autoplay，则忽略该属性
src	url（地址）	要播放的音频的 URL 地址

另外，<audio>标记可以通过 source 属性添加多个音频文件。具体格式如下：

```
<audio controls="controls">
<source src="123.ogg" type="audio/ogg">
<source src="123.mp3" type="audio/mpeg">
```

```
</audio>
```

6.1.3　音频解码器

音频解码器定义了音频数据流编码和解码的算法。其中，编码器主要是对数据流进行编码操作，用于存储和传输；解码器主要是对音频文件进行解码，然后进行播放操作。目前，使用较多的音频解码器是 Vorbis 和 ACC。

6.1.4　<audio>标记浏览器的支持情况

不同的浏览器对<audio> 标记支持也不同。表 6-2 列出了应用比较广泛的浏览器对<audio>标记的支持情况。

表 6-2　<audio>标记的浏览器支持情况

浏览器 / 音频格式	Firefox 3.5 及更高版本	IE 9.0 及更高版本	Opera 10.5 及更高版本	Chrome 3.0 及更高版本	Safari 3.0 及更高版本
Ogg Vorbis	支持		支持	支持	
MP3		支持		支持	支持
Wav	支持		支持		支持

6.2　<video>标记

与音频文件播放方式一样，大多数视频文件在网页上也是通过插件来播放的，常见的播放插件 Flash。由于不是所有的浏览器都拥有同样的插件，所以需要一种统一的包含视频的标准方法。与 HTML4 相比，HTML5 新增了<video>标记。

6.2.1　<video>标记概述

video 标记主要是定义播放视频文件或视频流的标准。支持 3 种视频格式，分别为 Ogg、WebM 和 MPEG4。

如果需要在 HTML5 网页中播放视频，输入的基本格式如下：

```
<video src="123.mp4" controls="controls"></ video >
```

另外，在< video >与</ video >之间插入的内容是供不支持 video 元素的浏览器显示的。

【例 6.2】（实例文件：ch06\6.2.html）

```
<!DOCTYPE html>
<html>
```

```
<head>
<title>video</title>
<head>
<body>
<video src="123.mp4" controls="controls">
您的浏览器不支持video标记!
</video>
</body>
</html>
```

如果用户的浏览器是 IE 11.0 之前的版本，则浏览效果如图 6-3 所示，可见 IE 11.0 之前的版本浏览器不支持<video>标记。

在 IE 11.0 中浏览效果如图 6-4 所示，可以看到加载的视频控制条界面。单击【播放】按钮，即可查看视频的内容。

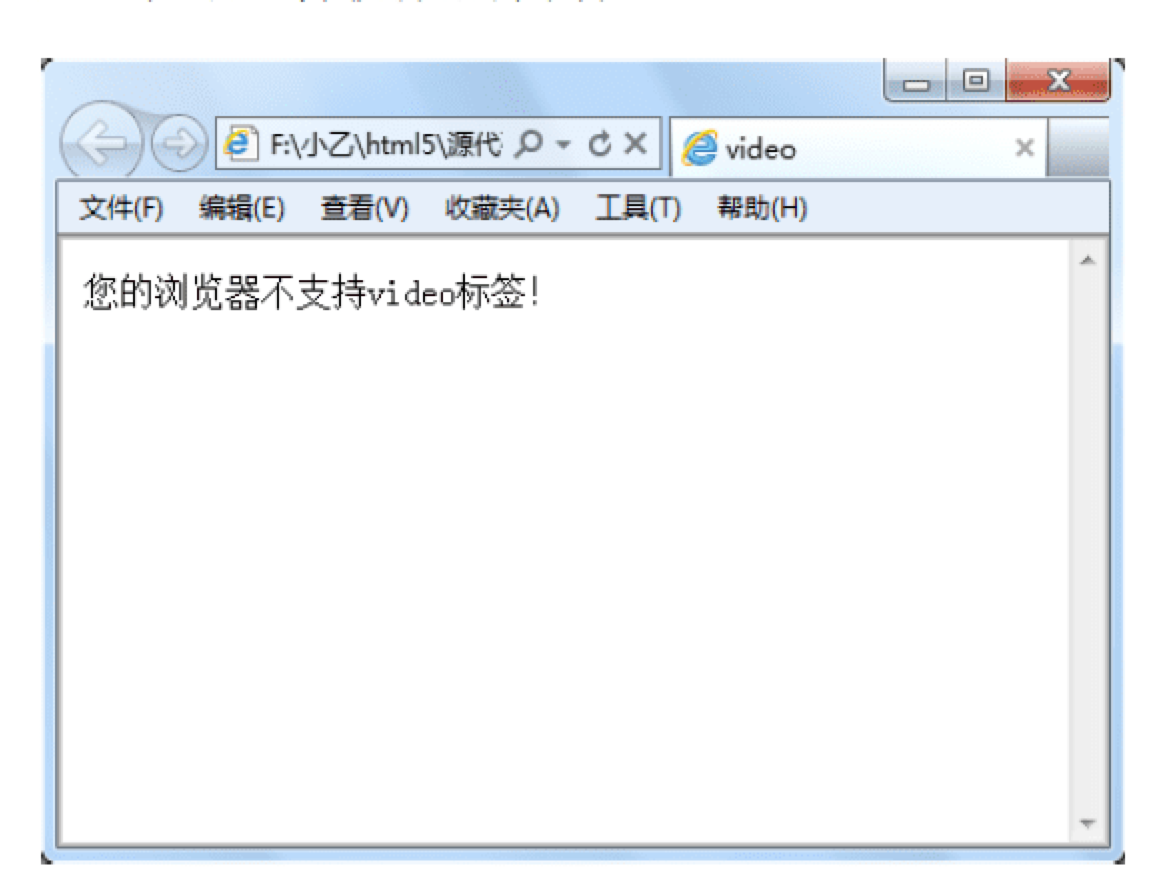


图 6-3　不支持<video>标记的效果



图 6-4　支持<video>标记的效果

6.2.2　<video>标记的属性

<video>标记的常见属性和含义如表 6-3 所示。

表 6-3　<video>标记常见属性

属性	值	描述
autoplay	autoplay	如果出现该属性，则视频在就绪后马上播放
controls	controls	如果出现该属性，则向用户显示控件，如播放按钮
loop	loop	如果出现该属性，则每当视频结束时重新开始播放
preload	preload	如果出现该属性，则视频在页面加载时进行加载，并预备播放 如果使用 "autoplay"，则忽略该属性
src	url	要播放的视频的 URL

（续表）

属性	值	描述
width	宽度值	设置视频播放器的宽度
height	高度值	设置视频播放器的高度
poster	url	当视频未响应或缓冲不足时，该属性值链接到一个图像。该图像将以一定的比例被显示出来

由上表可知，用户可以自定义视频文件显示的大小。例如，如果想让视频以 320×240 像素大小显示，就可以加入 width 和 height 属性。具体格式如下：

```
<video width="320" height="240" controls src="123.mp4" ></video>
```

另外，<video>标记可以通过 source 属性添加多个视频文件，具体格式如下：

```
<video controls="controls">
<source src="123.ogg" type="video/ogg">
<source src="123.mp4" type="video/mp4">
</video>
```

6.2.3　视频解码器

视频解码器定义了视频数据流编码和解码的算法。其中，编码器主要是对数据流进行编码操作，用于存储和传输；解码器主要是先对视频文件进行解码，然后进行播放操作。

目前，在 HTML5 中，使用比较多的视频解码文件是 Theora、H.264 和 VP8。

6.2.4　<video>标记浏览器的支持情况

不同的浏览器对<video>标记支持也不同。表 6-4 列出了应用比较广泛的浏览器对<video>标记的支持情况。

表 6-4　<video>标记的浏览器支持情况

浏览器 / 视频格式	Firefox 4.0 及更高版本	IE 9.0 及更高版本	Opera 10.6 及更高版本	Chrome 6.0 及更高版本	Safari 3.0 及更高版本
Ogg	支持		支持	支持	
MPEG 4		支持		支持	支持
WebM	支持		支持	支持	

6.3 音频和视频中的方法

在 HTML5 网页中，操作音频或视频文件的常用方法包括 canPlayType()方法、load()方法、play()方法和 pause()方法。

6.3.1 canPlayType()方法

canPlayType()方法用于检测浏览器是否能播放指定的音频或视频类型。canPlayType()方法返回值包含如下：

（1）probably：浏览器全面支持指定的音频或视频类型。

（2）maybe：浏览器可能支持指定的音频或视频类型。

（3）""（空字符串）：浏览器不支持指定的音频或视频类型。

注意，目前所有主流浏览器都支持 canPlayType()方法。Internet Explorer 8 及之前的版本不支持该方法。

【例 6.3】（实例文件：ch06\6.3.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <title>canPlayType()方法</title>
    </head>
    <body>
    <p>浏览器可以播放 MP4视频吗?<span>
    <button onclick="supportType(event,'video/mp4','avc1.42E01E,
mp4a.40.2')" type="button">检查</button>
    </span></p>
    <p>浏览器可以播放 OGG 音频吗?<span>
    <button onclick="supportType(event,'audio/ogg','theora, vorbis')"
type="button">检查</button>
    </span></p>
    <script>
    function supportType(e,vidType,codType)
    {
      myVid=document.createElement('video');
      isSupp=myVid.canPlayType(vidType+';codecs="'+codType+'"');
      if (isSupp=="")
      {
        isSupp="不支持";
      }
      e.target.parentNode.innerHTML="检查结果: " + isSupp;
    }
    </script>
```

```
</body>
</html>
```

在 IE 11.0 中浏览效果如图 6-5 所示。单击【检查】按钮，即可查看浏览器对音频和视频的支持情况，如图 6-6 所示。

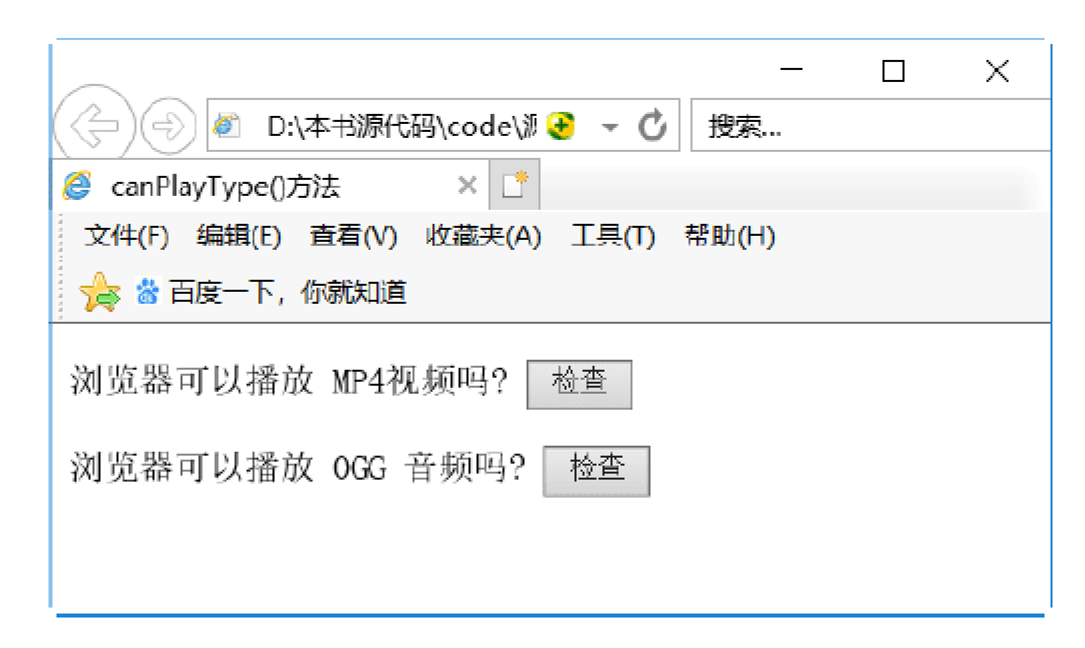


图 6-5　预览效果

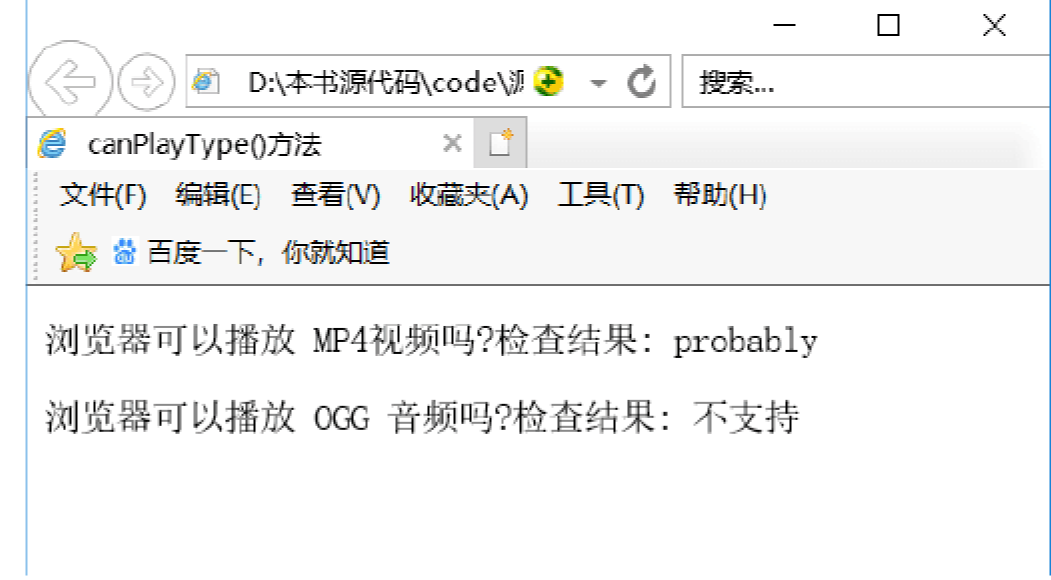


图 6-6　查看浏览器对音频和视频的支持情况

6.3.2　load()方法

load()方法用于重新加载音频或视频文件。load()方法的语法格式如下：

```
audio|video.load()
```

【例 6.4】（实例文件：ch06\6.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>load()方法</title>
</head>
<body>
<button onclick="changeSource()" type="button">更改加载视频</button>
<br />
<video id="video1" controls="controls" autoplay="autoplay">
  <source id="mp4 src" src="123.mp4" type="video/mp4">
  <source id="mp4 src" src="124.mp4" type="video/mp4">
  您的浏览器不支持 HTML5 video  标签。
</video>
<script>
function changeSource()
{
  document.getElementById("mp4 src").src="movie.mp4";
  document.getElementById("mp4 src").src="movie.mp4";
  document.getElementById("video1").load();
}
</script>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 6-7 所示。单击【更改加载视频】按钮，即可重新加载视频文件，如图 6-8 所示。



图 6-7　预览效果

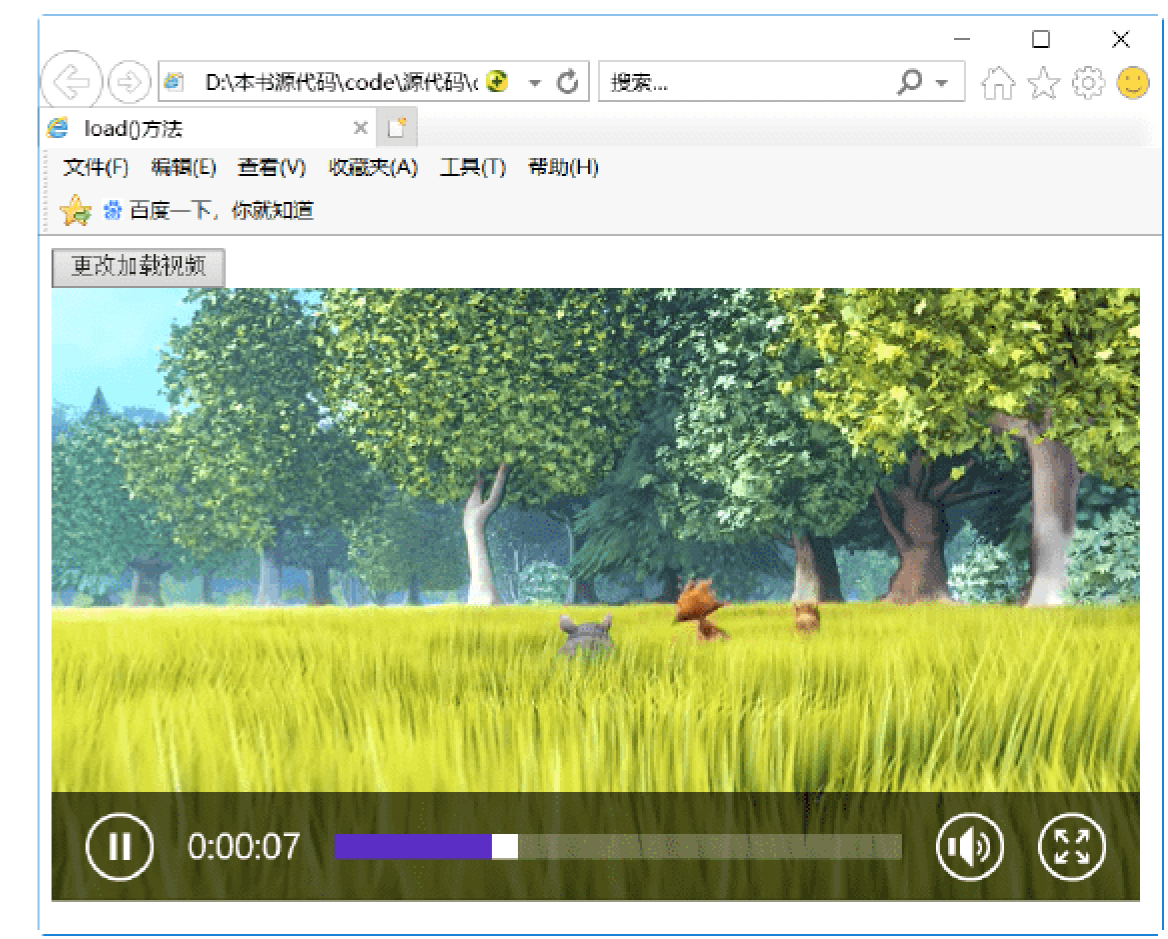


图 6-8　重新加载视频文件

6.3.3　play()方法和 pause()方法

play()方法用于开始播放音频或视频文件。pause()方法用于暂停当前播放的音频或视频文件。

【例 6.5】（实例文件：ch06\6.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title> play()方法</title>
</head>
<body>
<button onclick="playVid()" type="button">播放视频</button>
<button onclick="pauseVid()" type="button">暂停视频</button>
<br />
<video id="video1">
  <source src="124.mp4" type="video/mp4">
  您的浏览器不支持 HTML5 video  标签。
</video>
<script>
var myVideo=document.getElementById("video1");
function playVid()
{
  myVideo.play();
}

function pauseVid()
{
  myVideo.pause();
}
</script>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 6-9 所示。单击【播放视频】按钮，则视频开始播放；单击【暂停视频】按钮，则视频暂停播放。

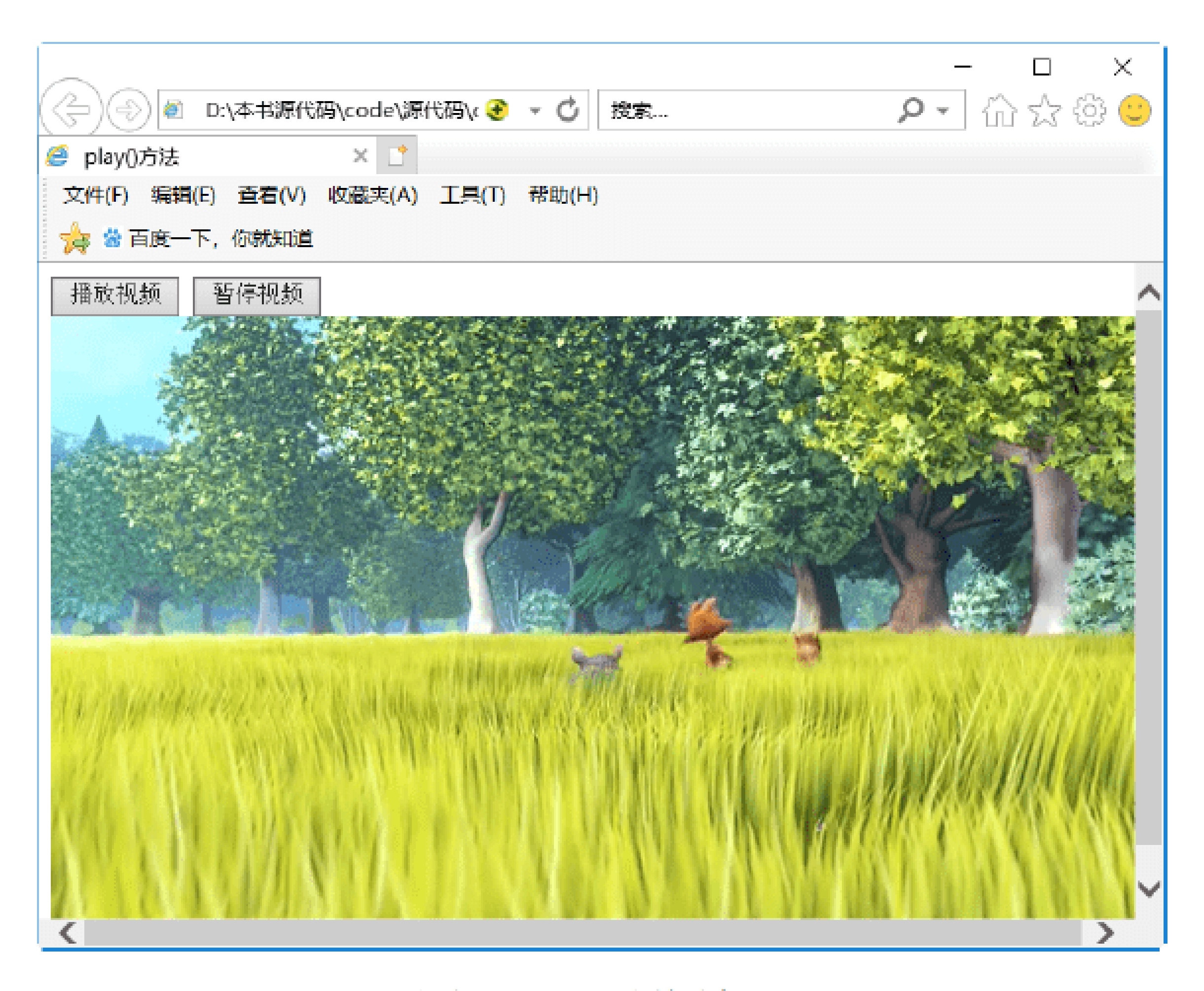


图 6-9 预览效果

6.4 音频和视频中的属性

在 HTML5 网页中，关于音频和视频的属性非常多，本节将挑选几个常用的进行讲解。

6.4.1 autoplay 属性

autoplay 属性设置与返回音频或视频是否在加载后立即开始播放。

设置 autoplay 属性的语法格式如下：

```
audio|video.autoplay=true|false
```

返回 autoplay 属性的语法格式如下：

```
audio|video.autoplay
```

其中，autoplay 属性的取值包括 true 和 false。

（1）true：设置音频或视频在加载后立即开始播放。

（2）false：默认值，设置音频或视频在加载后不立即开始播放。

【例 6.6】（实例文件：ch06\6.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title> autoplay属性</title>
</head>
<body>
<button onclick="enableAutoplay()" type="button">启动自动播放</button>
<button onclick="disableAutoplay()" type="button">禁用自动播放</button>
<button onclick="checkAutoplay()" type="button">检查自动播放状态</button>
```

```
    <br />
    <video id="video1" controls="controls">
      <source src="mov bbb.mp4" type="video/mp4">
      您的浏览器不支持 HTML5 video标签。
    </video>
    <script>
    myVid=document.getElementById("video1");
    function enableAutoplay()
    {
      myVid.autoplay=true;
      myVid.load();
    }
    function disableAutoplay()
    {
      myVid.autoplay=false;
      myVid.load();
    }
    function checkAutoplay()
    {
      alert(myVid.autoplay);
    }
    </script>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 6-10 所示。单击【启动自动播放】按钮，然后单击【检查自动播放状态】按钮，即可看到此时 autoplay 属性为 true。

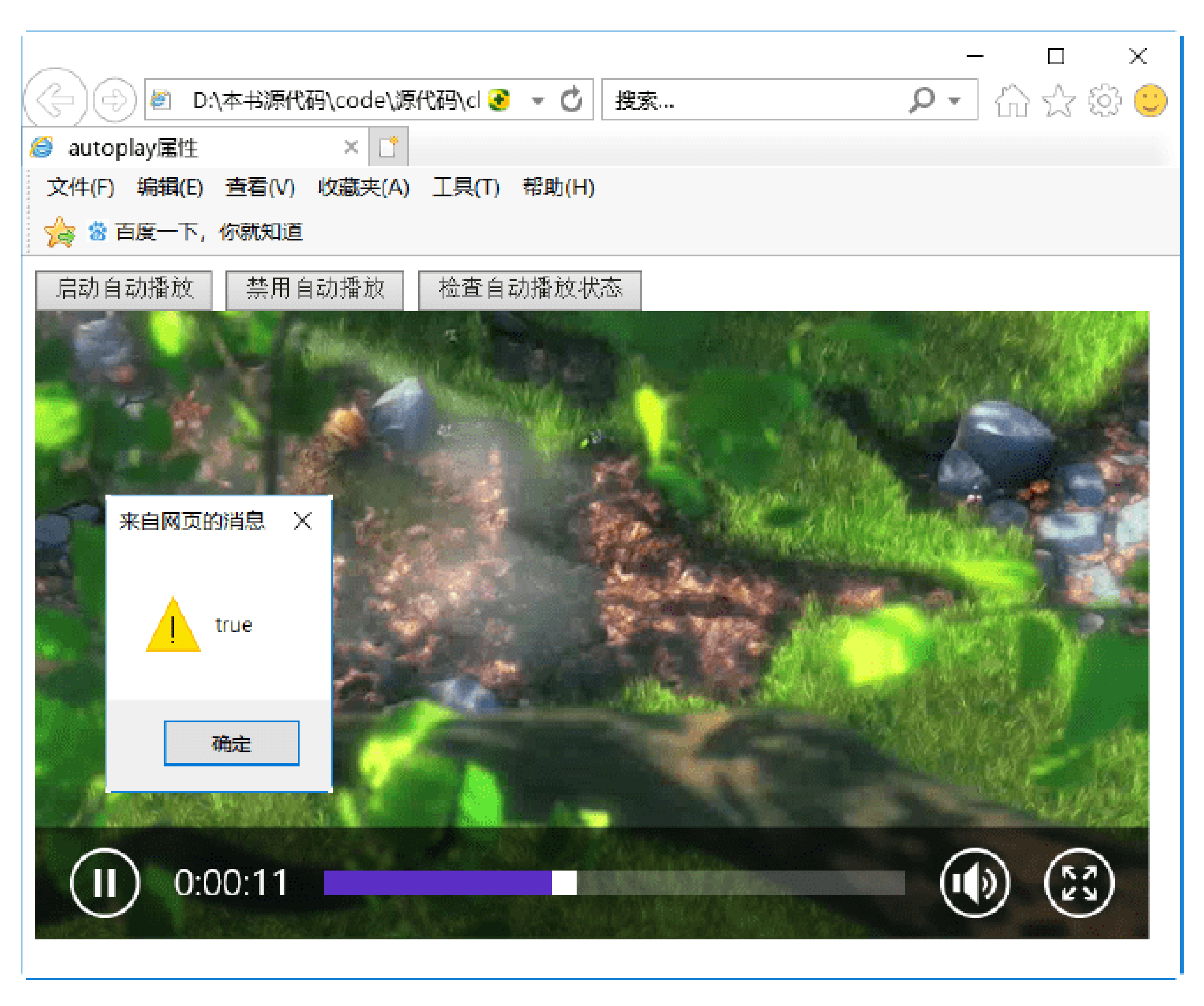


图 6-10　预览效果

6.4.2　buffered 属性

buffered 属性返回 TimeRanges 对象。TimeRanges 对象表示用户的音频或视频缓冲范围。缓冲范围指的是已缓冲音频或视频的时间范围。如果用户在音频或视频中跳跃播放，就会得到

多个缓冲范围。

返回 buffered 属性的语法格式如下：

```
audio|video.buffered
```

【例 6.7】（实例文件：ch06\6.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title> buffered 属性</title>
</head>
<body>
<button onclick="getFirstBuffRange()" type="button">获得视频的第一段缓冲范围
</button>
<br />
<video id="video1" controls="controls">
  <source src="mov bbb.mp4" type="video/mp4">
  您的浏览器不支持 HTML5 video  标签</video>
<script>
myVid=document.getElementById("video1");
function getFirstBuffRange()
{
  alert("开始: " + myVid.buffered.start(0) + "结束: "  +
myVid.buffered.end(0));
}
</script>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 6-11 所示。视频播放一段后，单击【获得视频的第一段缓冲范围】按钮，即可看到此时视频的缓冲范围。

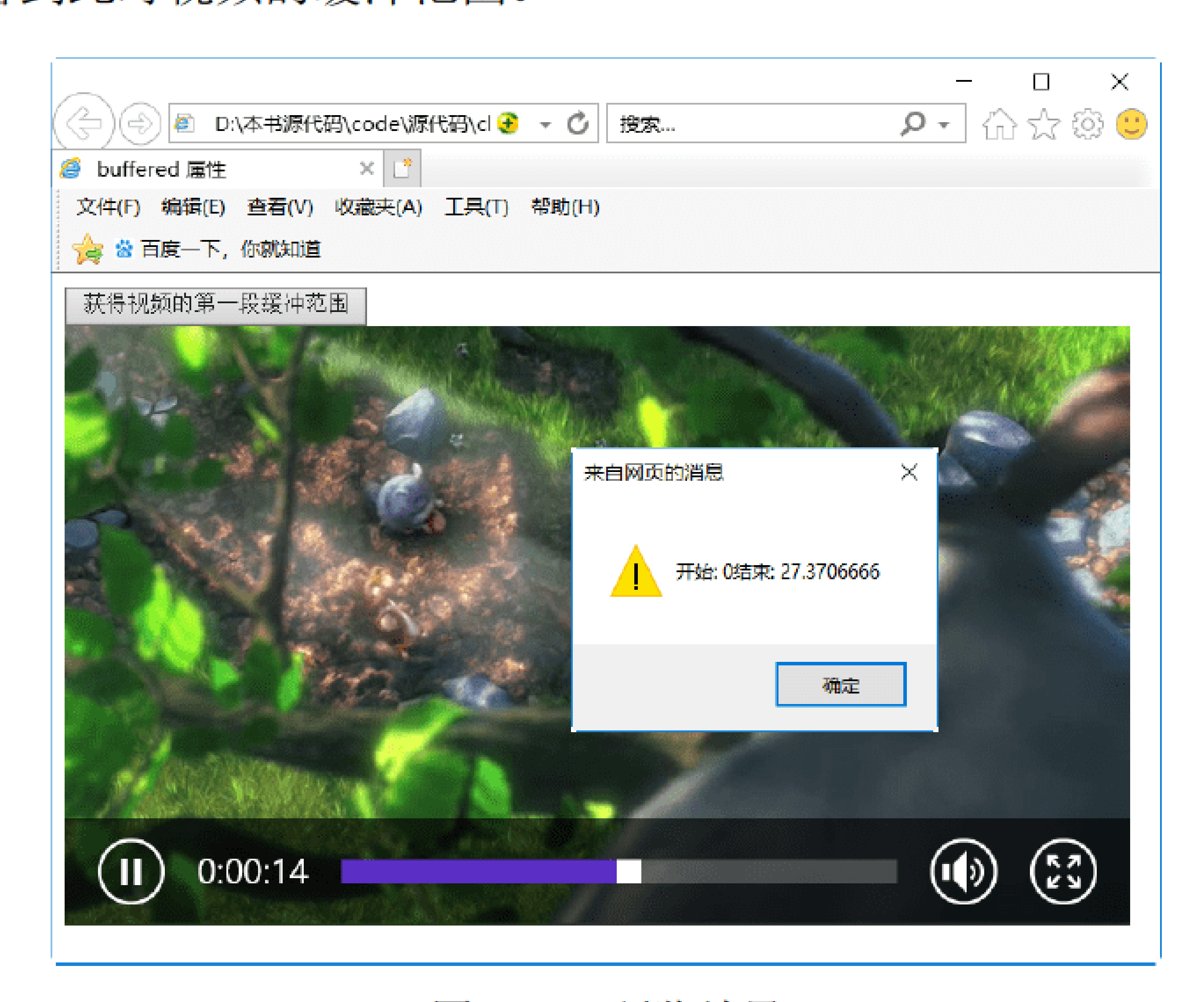


图 6-11 预览效果

6.4.3 controls 属性

controls 属性设置或返回浏览器应当显示标准的音频或视频控件。标准的音频或视频控件包括播放、暂停、进度条、音量、全屏切换、字幕和轨道。

设置 controls 属性的语法格式如下：

```
audio|video.controls=true|false
```

返回 controls 属性的语法格式如下：

```
audio|video.controls
```

其中，controls 属性的取值包括 true 和 false。

（1）true：设置显示控件。

（2）false：默认值，设置不显示控件。

【例 6.8】（实例文件：ch06\6.8.html）

```
<!DOCTYPE html>
<html>
<head>
<title>controls 属性</title>
</head>
<body>
<button onclick="enableControls()" type="button">启动控件</button>
<button onclick="disableControls()" type="button">禁用控件</button>
<button onclick="checkControls()" type="button">检查控件状态</button><br />
<video id="video1">
  <source src="124.mp4" type="video/mp4">
  您的浏览器不支持 HTML5 video 标签。</video>
<script>
myVid=document.getElementById("video1");
function enableControls()
{
  myVid.controls=true;
  myVid.load();
}
function disableControls()
{
  myVid.controls=false;
  myVid.load();
}
function checkControls()
{
  alert(myVid.controls);
}
```

```
</script>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 6-12 所示。单击【启动控件】按钮，然后单击【检查控件状态】按钮，即可看到此时 controls 属性为 true。

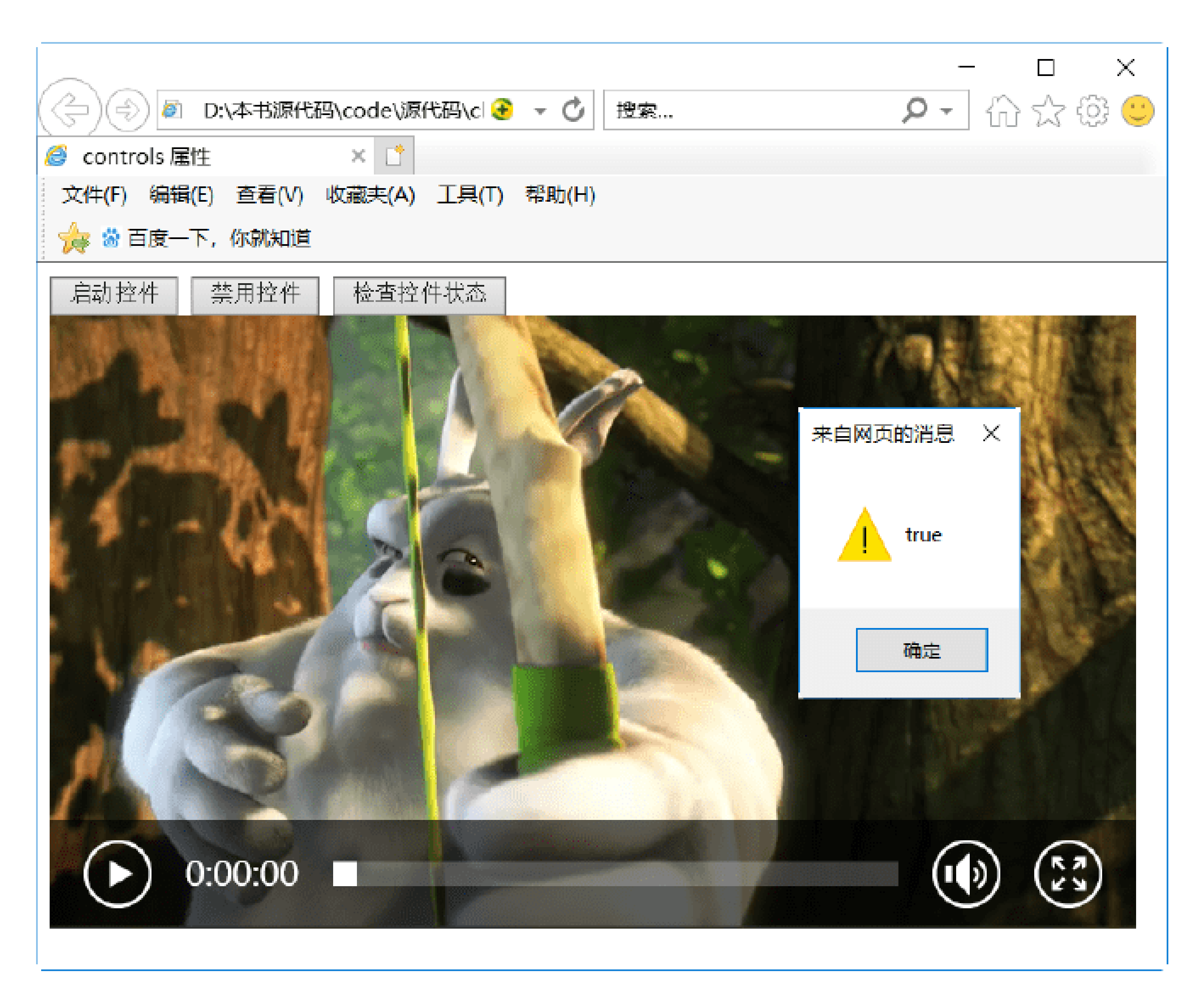


图 6-12　预览效果

6.4.4　currentSrc 属性

currentSrc 属性返回当前音频或视频的 URL。如果未设置音频或视频，则返回空字符串。

返回 currentSrc 属性的语法格式如下：

```
audio|video.currentSrc
```

【例 6.9】（实例文件：ch06\6.9.html）

```
<!DOCTYPE html>
<html>
<head>
<title> currentSrc属性</title>
</head>
<body>
<button onclick="getVid()" type="button">获得当前视频的URL</button><br />
<video id="video1" controls="controls">
  <source src="124.mp4" type="video/mp4">
  您的浏览器不支持 HTML5 video 标签。</video>
<script>
myVid=document.getElementById("video1");
function getVid()
```

```
    {
      alert(myVid.currentSrc);
    }
</script>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 6-13 所示。单击【获得当前视频的 URL】按钮，即可看到当前视频的 URL 路径。

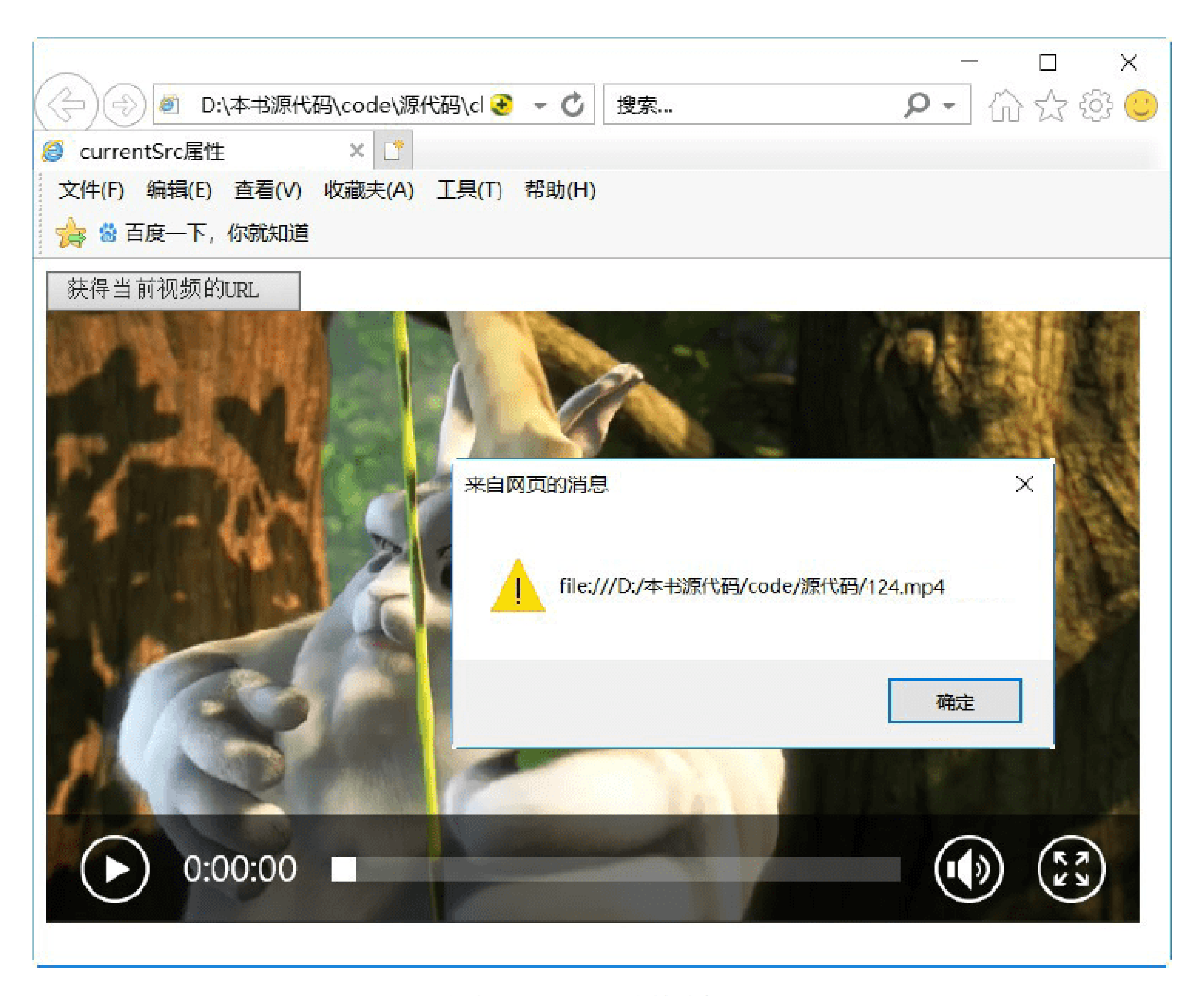


图 6-13　预览效果

6.5　专家解惑

1. 在 HTML5 网页中添加所支持格式的视频，不能在 Firefox 8.0 浏览器中正常播放，为什么？

目前，HTML5 的<video>标记对视频的支持，不仅仅有视频格式的限制，还有对解码器的限制。规定如下：

（1）如果视频是 Ogg 格式的文件，则需要带有 Thedora 视频编码和 Vorbis 音频编码的视频。

（2）如果视频是 MPEG4 格式的文件，则需要带有 H.264 视频编码和 AAC 音频编码的视频。

（3）如果视频是 WebM 格式的文件，则需要带有 VP8 视频编码和 Vorbis 音频编码的视频。

2. 在 HTML5 网页中添加 mp4 格式的视频文件，为什么在不同的浏览器中视频控件显示的外观不同？

在 HTML5 中规定 controls 属性用于视频文件的播放、暂停、停止和调节音量的操作。因为 Controls 是一个布尔属性，所以可以赋予任何值。一旦添加了此属性，就等于告诉浏览器需要显示播放控件并允许用户操作。因为每一个浏览器负责内置视频控件的外观不同，所以在不同的浏览器中将显示不同的视频控件外观。

第7章　HTML5绘制图形

HTML5 呈现了很多的新特性，这在之前的 HTML 中是见不到的，其中一个最值得提及的特性就是 HTML 的<canvas>标记，可以对 2D 或位图进行动态、脚本的渲染。Canvas 是一个矩形区域，使用 JavaScript 可以控制其每一个像素。

7.1　canvas 概述

Canvas 是一个新的 HTML 元素，可以被 Script 语言（通常是 JavaScript）用来绘制图形。

7.1.1　添加 canvas 元素

<canvas>标记是一个矩形区域，包含两个属性 width 和 height，分别表示矩形区域的宽度和高度。这两个属性都是可选的，并且都可以通过 CSS 来定义，其默认值是 300 像素和 150 像素。

Canvas 在网页中常用的形式如下：

```
    <canvas id="myCanvas" width="300" height="200" style="border:1px solid
#c3c3c3;">
    Your browser does not support the canvas element.
    </canvas>
```

上面示例代码中，id 表示画布对象名称；width 和 height 分别表示宽度和高度。最初的画布是不可见的，此处为了观察该矩形区域，使用了 CSS 样式，即<style>标记，Style 表示画布的样式。如果浏览器不支持画布标记，就会显示画布中间的提示信息。

画布 canvas 本身不具有绘制图形的功能，只是一个容器。如果读者对于 Java 语言有所了解，就会发现 HTML5 的画布和 Java 中的 Panel 面板非常相似，都可以在容器绘制图形。既然 canvas 画布元素放好了，就可以使用脚本语言 JavaScript 在网页上绘制图形了。

使用 canvas 结合 JavaScript 绘制图形，一般需要下面几个步骤。

01 JavaScript 使用 id 来寻找 canvas 元素，即获取当前画布对象。

```
var c=document.getElementById("myCanvas");
```

02 创建 context 对象。

```
var cxt=c.getContext("2d");
```

getContext 函数返回一个指定 contextId 的上下文对象，如果指定的 id 不被支持，则返回 null，当前唯一被强制必须支持的是 2D（也许在将来会有 3D）。注意，指定的 id 对大小写是非常敏感的。对象 cxt 建立之后，就可以拥有多种绘制路径、矩形、圆形、字符及添加图像的方法。

03 绘制图形。

```
cxt.fillStyle="#FF0000";
cxt.fillRect(0,0,150,75);
```

fillStyle 函数将其染成红色。fillRect 函数规定了形状、位置和尺寸，这两行代码绘制一个红色的矩形。

7.1.2　绘制矩形

单独的<canvas>标记只是在页面中定义了一个矩形区域，并无特别之处，开发人员只有配合使用 JavaScript 脚本，才能完成各种图形、线条及复杂图形的变换操作。与基于 SVG 来实现同样绘图效果进行比较，canvas 绘图是一种像素级别的位图绘图技术，SVG 则是一种矢量绘图技术。

使用 canvas 和 JavaScript 绘制一个矩形，可能会涉及一个或多个函数，这些函数如表 7-1 所示。

表 7-1　绘制函数

函数	功能
fillRect	绘制一个矩形，该矩形区域没有边框，只有填充色。这个函数有 4 个参数，前两个表示左上角的坐标位置，第 3 个参数为长度，第 4 个参数为高度
strokeRect	绘制一个带边框的矩形。该函数的 4 个参数的解释同上
clearRect	清除一个矩形区域，被清除的区域没有任何线条。该函数的 4 个参数的解释同上

【例 7.1】（实例文件：ch07\7.1.html）

```
<!DOCTYPE html>
<html>
<body>
```

```
    <canvas id="myCanvas" width="300" height="200" style="border:1px solid
blue">Your browser does not support the canvas element</canvas>
    <script type="text/javascript">
    var c=document.getElementById("myCanvas");
    var cxt=c.getContext("2d");
    cxt.fillStyle="rgb(0,0,200)";
    cxt.fillRect(10,20,100,100);
    </script>
    </body>
    </html>
```

在上面的代码中，首先定义了一个画布对象，其 id 名称为 myCanvas，高度和宽度分别为 200 像素和 300 像素，并定义了画布的边框显示样式。在 JavaScript 代码中，首先获取画布对象，然后使用 getContext 获取当前 2d 的上下文对象，并使用 fillRect 绘制一个矩形。其中涉及一个 fillStyle 属性，用于设置填充的颜色、透明度等，如果设置为“rgb(200,0,0)”，则表示一个颜色，不透明；如果设为“rgba(0,0,200,0.5)”，则表示一个颜色，透明度为 50%。

在 IE 11.0 中浏览效果如图 7-1 所示，可以看到网页中，在一个蓝色边框中显示了一个蓝色矩形。

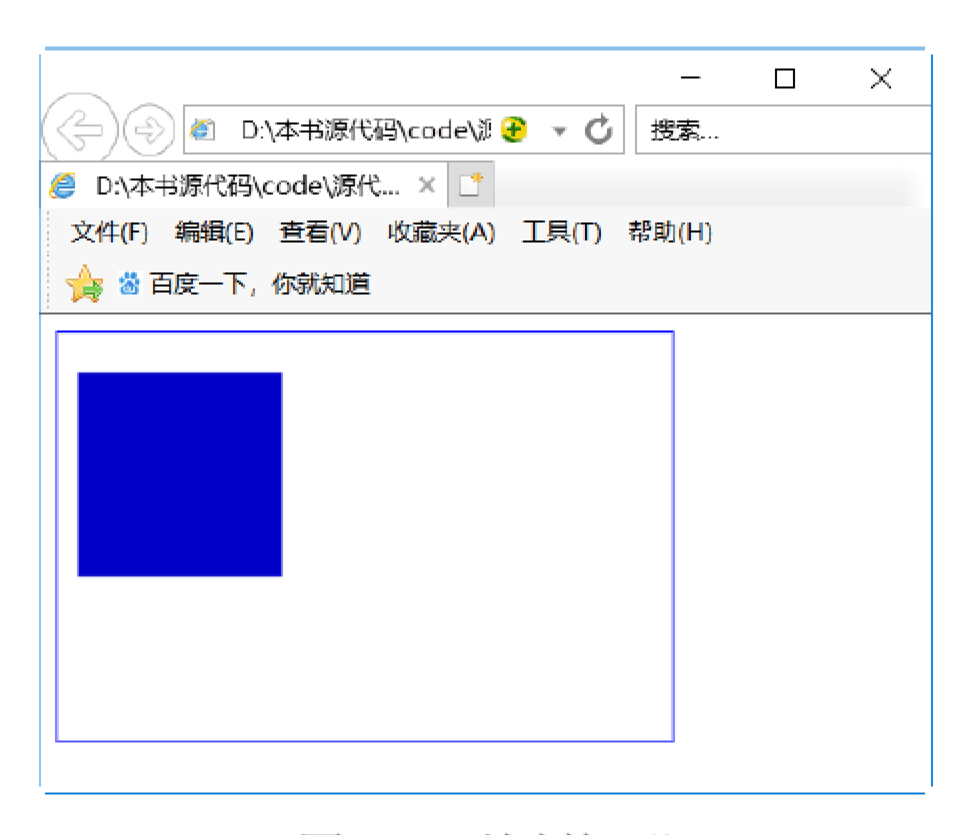


图 7-1　绘制矩形

7.2　绘制基本形状

画布 canvas 结合 JavaScript 不但可以绘制简单的矩形，还可以绘制一些其他的常见图形，如直线、圆等。

7.2.1　绘制圆形

基于 canvas 的绘图并不是直接在<canvas>标记所创建的绘图画面上进行各种绘图操作，而是依赖画面所提供的渲染上下文（Rendering Context），所有的绘图命令和属性都定义在渲染

上下文中。在通过 canvas id 获取相应的 DOM 对象之后，首先要做的事情就是获取渲染上下文对象。渲染上下文与 canvas 一一对应，无论对同一 canvas 对象调用几次 getContext()函数，都将返回同一个上下文对象。

在画布中绘制圆形，可能要涉及下面几个函数，如表 7-2 所示。

表 7-2　绘制函数

函数	功能
beginPath()	开始绘制路径
arc(x,y,radius,startAngle,endAngle,anticlockwise)	x 和 y 定义的是圆的原点，radius 是圆的半径，startAngle 和 endAngle 是弧度，不是度数，anticlockwise 用来定义画圆的方向，值是 true 或 false
closePath()	结束路径的绘制
fill()	进行填充
stroke()	方法设置边框

路径是绘制自定义图形的好方法，在 canvas 中通过 beginPath()函数开始绘制路径，这个时候就可以绘制直线、曲线等，绘制完成后调用 fill()和 stroke()函数完成填充和设置边框，最后通过 closePath()函数结束路径的绘制。

【例 7.2】（实例文件：ch07\7.2.html）

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="200" style="border:1px solid
blue">
Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.fillStyle="#FFaa00";
cxt.beginPath();
cxt.arc(70,18,15,0,Math.PI*2,true);
cxt.closePath();
cxt.fill();
</script>
</body>
</html>
```

在上面的 JavaScript 代码中，首先使用 beignPath 函数开启一个路径，然后绘制一个圆形，

最后关闭这个路径并填充。

在 IE 11.0 中浏览效果如图 7-2 所示，可以看到网页中，在矩形边框中显示了一个黄色的圆。

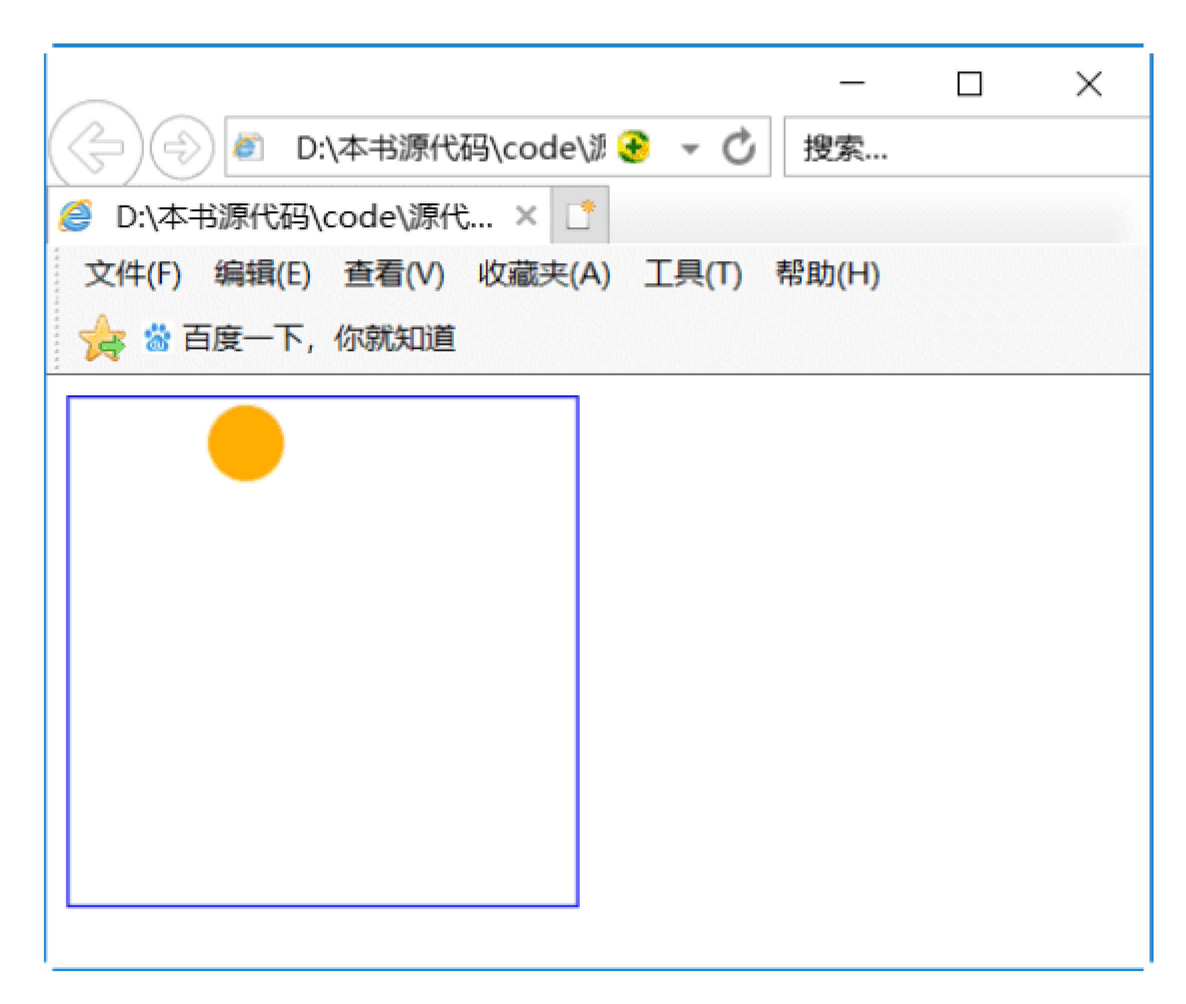


图 7-2　绘制椭圆

7.2.2　使用 moveTo 与 lineTo 绘制直线

在每个 canvas 实例对象中都拥有一个 path 对象，创建自定义图形的过程就是不断对 path 对象操作的过程。每当开始一次新的图形绘制任务，都需要先使用 beginPath()函数来重置 path 对象至初始状态，进而通过一系列对 moveTo/lineTo 等画线函数的调用，绘制期望的路径。其中 moveTo(x, y)函数设置绘图起始坐标，而 lineTo(x,y)等画线函数可以从当前起点绘制直线、圆弧以及曲线到目标位置。最后一步，也是可选的步骤，是调用 closePath()函数将自定义图形进行闭合，该函数将自动创建一条从当前坐标到起始坐标的直线。

绘制直线常用的函数是 moveTo 和 lineTo，其含义如表 7-3 所示。

表 7-3　绘制函数

函数或属性	功能
moveTo(x,y)	不绘制，只是将当前位置移动到新目标坐标（x,y），并作为线条开始点
lineTo(x,y)	绘制线条到指定的目标坐标(x,y)，并且在两个坐标之间画一条直线。不管调用它们哪一个，都不会真正画出图形，因为还没有调用 stroke（绘制）和 fill（填充）函数。当前只是在定义路径的位置，以便后面绘制时使用
strokeStyle	属性是指定线条的颜色
lineWidth	属性设置线条的粗细

【例 7.3】（实例文件：ch07\7.3.html）

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="200" style="border:1px solid blue">
Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.beginPath();
cxt.strokeStyle="rgb(0,182,0)";
cxt.moveTo(10,10);
cxt.lineTo(150,50);
cxt.lineTo(10,50);
cxt.lineWidth=14;
cxt.stroke();
cxt.closePath();
</script>
</body>
</html>
```

上面代码中，使用 moveTo 函数定义一个坐标位置为（10,10），下面以此坐标位置为起点绘制两条不同的直线，并使用 lineWidth 设置直线的宽度，使用 strokeStyle 设置直线的颜色，使用 lineTo 设置两条不同直线的结束位置。

在 IE 11.0 中浏览效果如图 7-3 所示，可以看到网页中绘制了两条直线，这两条直线在某一点交叉。

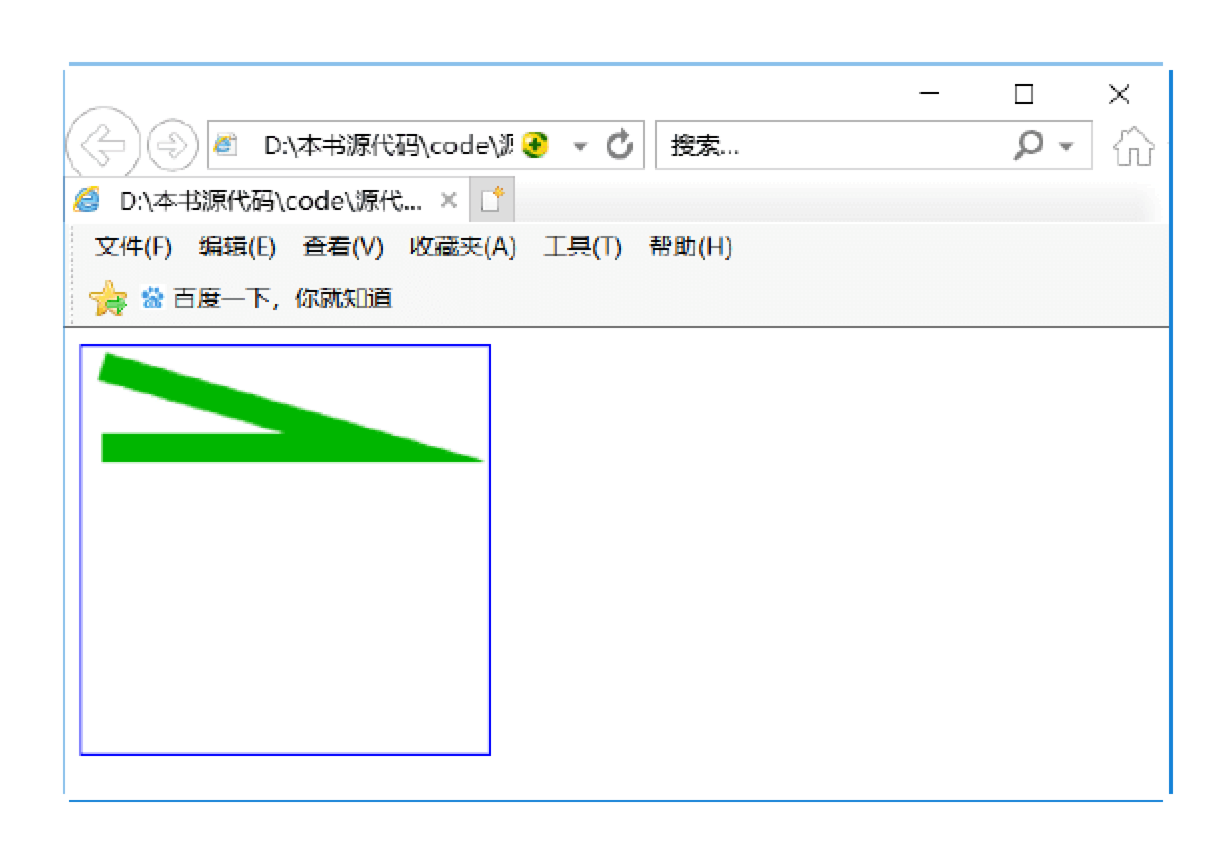


图 7-3　绘制直线

7.2.3　使用 bezierCurveTo 绘制贝济埃曲线

在数学的数值分析领域中，贝济埃曲线（Bézier 曲线）是电脑图形学中相当重要的参数曲

线。更高维度的广泛化贝济埃曲线就称作贝济埃曲面，其中贝济埃三角是一种特殊的实例。

bezierCurveTo()表示为一个画布的当前子路径添加一条三次贝塞尔曲线。这条曲线的开始点是画布的当前点，结束点是 (x, y)。两条贝塞尔曲线控制点 (cpX1, cpY1) 和 (cpX2, cpY2) 定义了曲线的形状。这个方法返回时，当前位置为 (x, y)。

方法 bezierCurveTo 具体格式如下：

```
bezierCurveTo(cpX1, cpY1, cpX2, cpY2, x, y)
```

其参数的含义如表 7-4 所示。

表 7-4　参数的含义

参数	描述
cpX1, cpY1	和曲线的开始点（当前位置）相关联的控制点的坐标
cpX2, cpY2	和曲线的结束点相关联的控制点的坐标
x, y	曲线的结束点的坐标

【例 7.4】（实例文件：ch07\7.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>贝济埃曲线</title>
<script>
    function draw(id)
    {
        var canvas=document.getElementById(id);
        if(canvas==null)
        return false;
        var context=canvas.getContext('2d');
        context.fillStyle="#eeeeff";
        context.fillRect(0,0,400,300);    //绘制一个矩形
        var n=0;
        var dx=150;
        var dy=150;
        var s=100;
        context.beginPath();
        context.globalCompositeOperation='and';
        context.fillStyle='rgb(100,255,100)';
        context.strokeStyle='rgb(0,0,100)';
        var x=Math.sin(0);
        var y=Math.cos(0);
        var dig=Math.PI/15*11;
        for(var i=0;i<30;i++)
        {
```

```
                var x=Math.sin(i*dig);
                var y=Math.cos(i*dig);

    context.bezierCurveTo(dx+x*s,dy+y*s-100,dx+x*s+100,dy+y*s,dx+x*s,dy+y*s
);
            }
            context.closePath();
            context.fill();
            context.stroke();
        }
    </script>
    </head>
    <body onload="draw('canvas');">
    <h1>绘制元素</h1>
    <canvas id="canvas" width="400" height="300" />
    </body>
    </html>
```

上面函数 draw 代码中，首先使用语句“fillRect(0,0,400,300)”绘制了一个矩形，其大小和画布相同，其填充颜色为浅青色。下面定义几个变量，用于设置曲线的坐标位置，在 for 循环中使用 bezierCurveTo 绘制贝济埃曲线。

在 IE 11.0 中浏览效果如图 7-4 所示，可以看到网页中显示了一个贝济埃曲线。

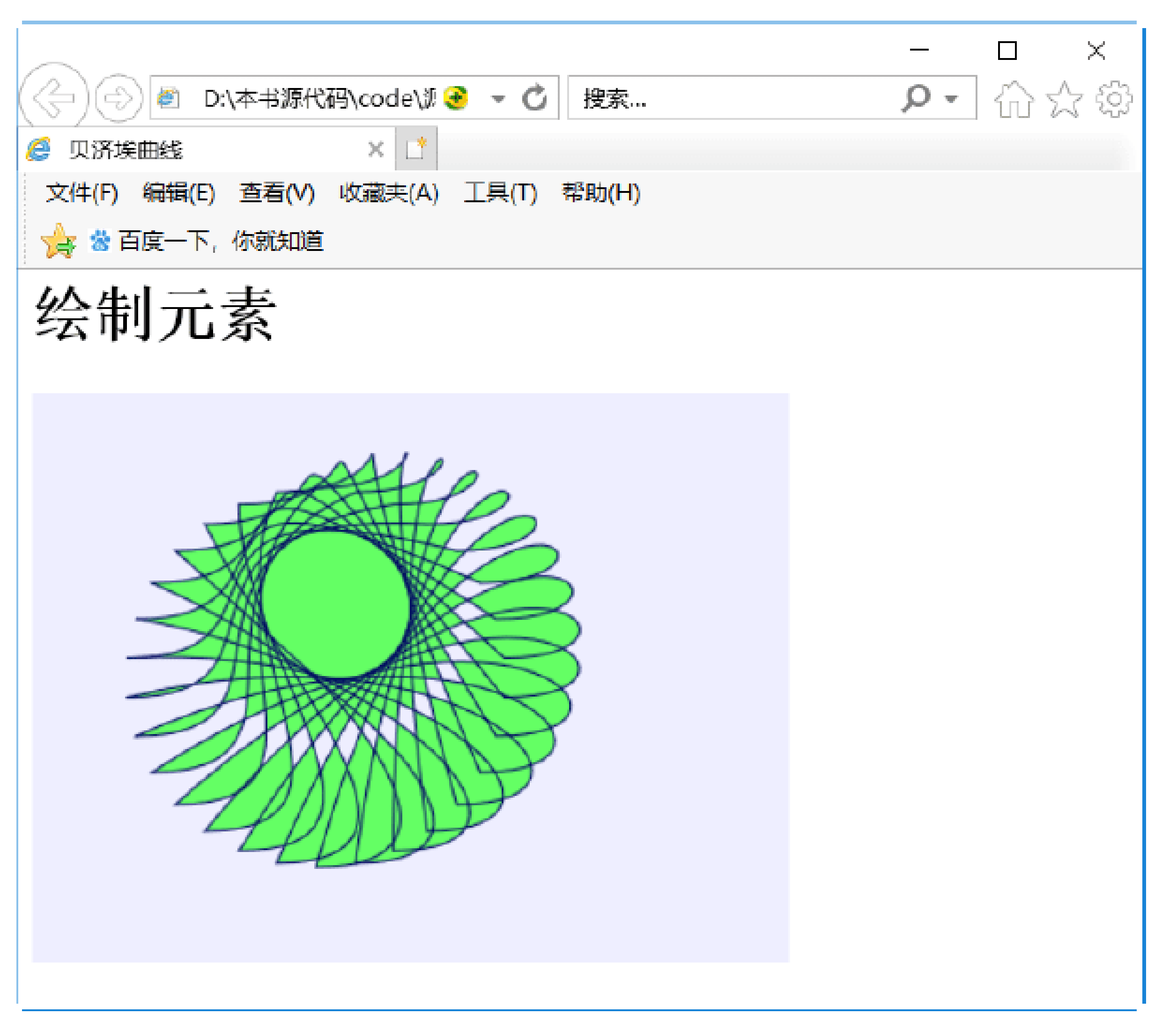


图 7-4　贝济埃曲线

7.3　绘制渐变图形

渐变是两种或更多颜色的平滑过渡，是指在颜色集上使用逐步抽样算法，并将结果应用于描边样式和填充样式中。Canvas 的绘图上下文支持两种类型的渐变：线性渐变和放射性渐变，

其中放射性渐变也称为径向渐变。

7.3.1 绘制线性渐变

创建一个简单的渐变非常容易，比使用 Photoshop 还要快，使用渐变需要以下 3 个步骤。

01 创建渐变对象。

```
var gradient=cxt.createLinearGradient(0,0,0,canvas.height);
```

02 为渐变对象设置颜色，指明过渡方式。

```
gradient.addColorStop(0,'#fff');
gradient.addColorStop(1,'#000');
```

03 在 context 上为填充样式或描边样式设置渐变。

```
cxt.fillStyle=gradient;
```

要设置显示颜色，在渐变对象上使用 addColorStop 函数即可。除了可以变换成其他颜色外，还可以为颜色设置 alpha 值（如透明），并且 alpha 值也是可以变化的。为了达到这样的效果，需要使用颜色值的另一种表示方法，例如内置 alpha 组件的 CSSrgba 函数。

绘制线性渐变，会使用到如表 7-5 所示的几个函数。

表 7-5　绘制函数

函数	功能
addColorStop	函数允许指定两个参数：颜色和偏移量。颜色参数是指开发人员希望在偏移位置描边或填充时所使用的颜色。偏移量是一个 0.0~1.0 之间的数值，代表沿着渐变线渐变的距离有多远
createLinearGradient(x0,y0,x1,x1)	沿着直线从（x0,y0）至（x1,y1）绘制渐变

【例 7.5】（实例文件：ch07\7.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>线性渐变</title>
</head>
<body>
<h1>绘制线性渐变</h1>
<canvas id="canvas" width="400" height="300" style="border:1px solid red"/>
<script type="text/javascript">
var c=document.getElementById("canvas");
```

```
var cxt=c.getContext("2d");    //使用2D环境对象产生了一个线性渐变对象
var gradient=cxt.createLinearGradient(0,0,0,canvas.height);
gradient.addColorStop(0,'#fff');  // 设置渐变颜色
gradient.addColorStop(1,'#000');  // 设置渐变颜色
cxt.fillStyle=gradient;           //将渐变填充到上下文环境的样式中
cxt.fillRect(0,0,400,400);
</script>
</body>
</html>
```

上面的代码使用 2D 环境对象产生了一个线性渐变对象，渐变的起始点是（0，0），渐变的结束点是（0，canvas.height），下面使用 addColorStop 函数设置渐变颜色，最后将渐变填充到上下文环境的样式中。

在 IE 11.0 中浏览效果如图 7-5 所示，可以看到网页中创建了一个垂直方向上的渐变，从上到下颜色逐渐变深。

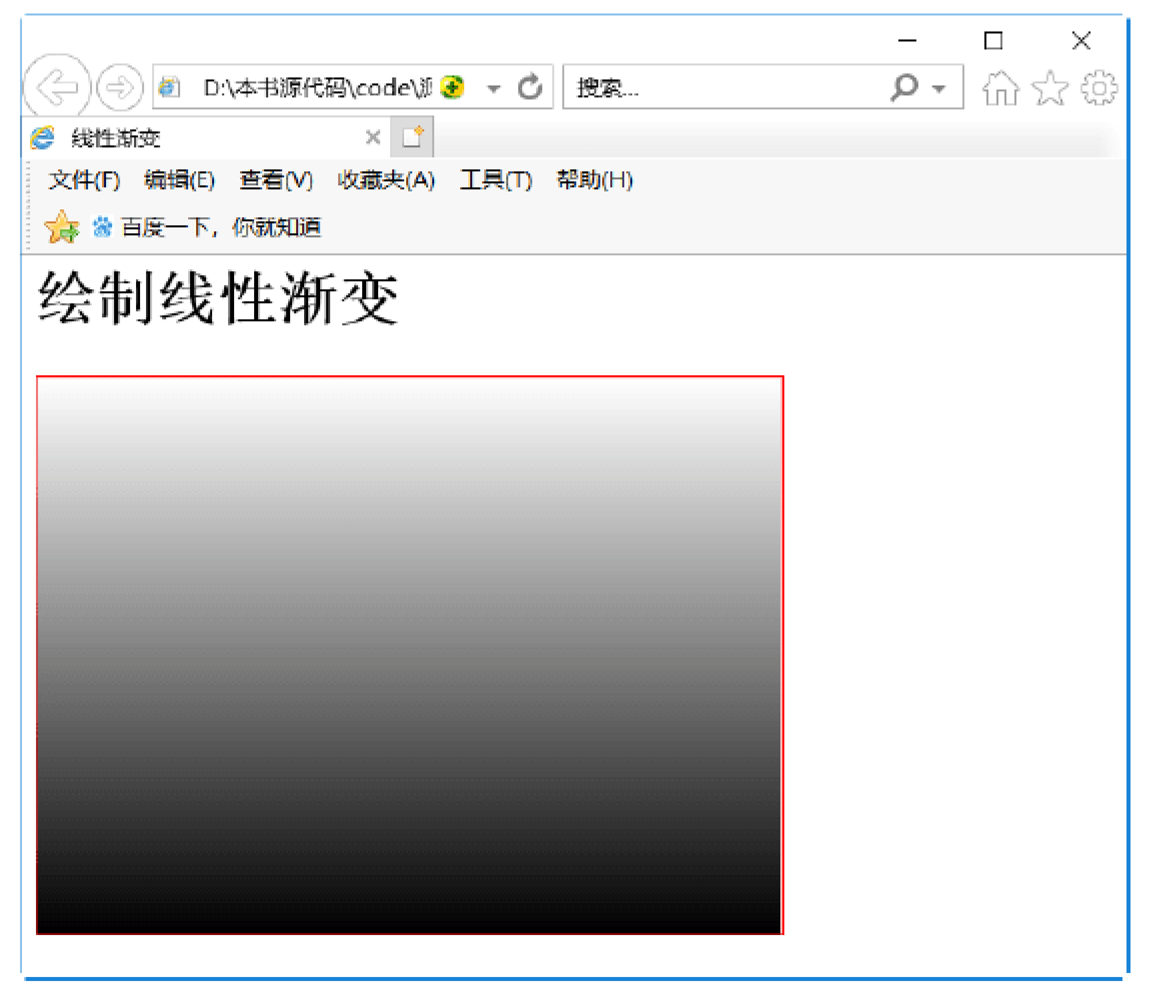


图 7-5　线性渐变

7.3.2　绘制径向渐变

除了线性渐变以外，HTML5 Canvas API 还支持放射性渐变，所谓放射性渐变就是颜色会介于两个指定圆间的锥形区域平滑变化。放射性渐变和线性渐变使用的颜色终止点是一样的。如果要实现放射线渐变，即径向渐变，需要使用函数 createRadialGradient。

createRadialGradient(x0,y0,r0,x1,y1,r1)函数表示沿着两个圆之间的锥面绘制渐变。其中前 3 个参数代表开始圆的圆心为（x0,y0），半径为 r0。最后 3 个参数代表结束圆的圆心为(x1,y1)，半径为 r1。

【例 7.6】（实例文件：ch07\7.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>径向渐变</title>
</head>
<body>
<h1>绘制径向渐变</h1>
<canvas id="canvas" width="400" height="300" style="border:1px solid red"/>
<script type="text/javascript">
var c=document.getElementById("canvas");
var cxt=c.getContext("2d");
var
gradient=cxt.createRadialGradient(canvas.width/2,canvas.height/2,0,canvas.w
idth/2,canvas.height/2,150);   //创建渐变对象gradient
gradient.addColorStop(0,'#fff');   //添加渐变颜色
gradient.addColorStop(1,'#000');   //添加渐变颜色
cxt.fillStyle=gradient;            //填充渐变颜色
cxt.fillRect(0,0,400,400);
</script>
</body>
</html>
```

上面代码中，首先创建渐变对象 gradient，此处使用方法 createRadialGradient 创建了一个径向渐变，下面使用 addColorStop 添加颜色，最后将渐变填充到上下文环境中。

在 IE 11.0 中浏览效果如图 7-6 所示，可以看到网页中，从圆的中心亮点开始，向外逐步发散，形成了一个径向渐变。

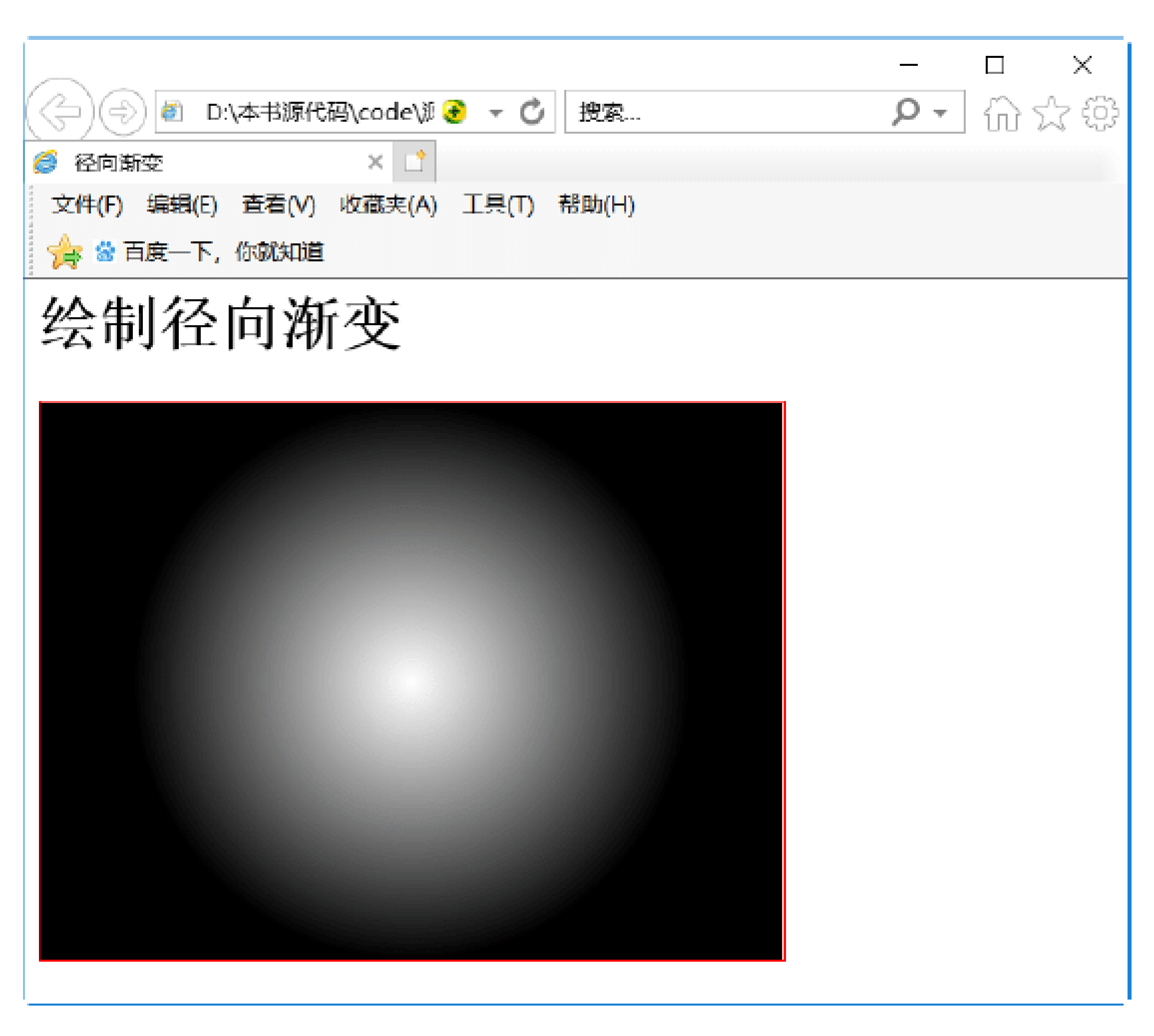


图 7-6　径向渐变

7.4　绘制变形图形

画布 canvas 不但可以使用 moveTo 这样的方法来移动画笔，绘制图形和线条，还可以使用变换来调整画笔下的画布。变换的方法包括旋转、缩放、变形、平移等。

7.4.1　变换原点坐标

平移（translate）即将绘图区相对于当前画布的左上角进行平移。如果不进行变形，则绘图区原点和画布原点是重叠的，绘图区相当于画图软件里的热区或当前层；如果进行变形，则坐标位置会移动到一个新位置。

如果要对图形实现平移，就需要使用函数 translate（x,y），该函数表示在平面上平移，即以原来原点为参考，然后以偏移后的位置作为坐标原点。也就是说，原来在（100,100），然后 translate（1,1）新的坐标原点在（101,101）而不是（1,1）。

【例 7.7】（实例文件：ch07\7.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title>绘制坐标变换</title>
<script>
    function draw(id)
    {
        var canvas=document.getElementById(id);
        if(canvas==null)
        return false;
        var context=canvas.getContext('2d');
        context.fillStyle="#eeeeff";
        context.fillRect(0,0,400,300);
        context.translate(200,50);
        context.fillStyle='rgba(255,0,0,0.25)';
        for(var i=0;i<50;i++){
            context.translate(25,25);
            context.fillRect(0,0,100,50);   }
    }
</script>
</head>
<body onload="draw('canvas');">
<h1>变换原点坐标</h1>
<canvas id="canvas" width="400" height="300" />
</body>
</html>
```

在 draw 函数中，使用 fillRect 函数绘制了一个矩形，然后使用 translate 函数平移到一个新

位置，并从新位置开始，使用 for 循环连续移动多次坐标原点，即多次绘制矩形。

在 IE 11.0 中浏览效果如图 7-7 所示，可以看到网页中从坐标位置（200,50）开始绘制矩形，并每次以指定的平移距离绘制矩形。

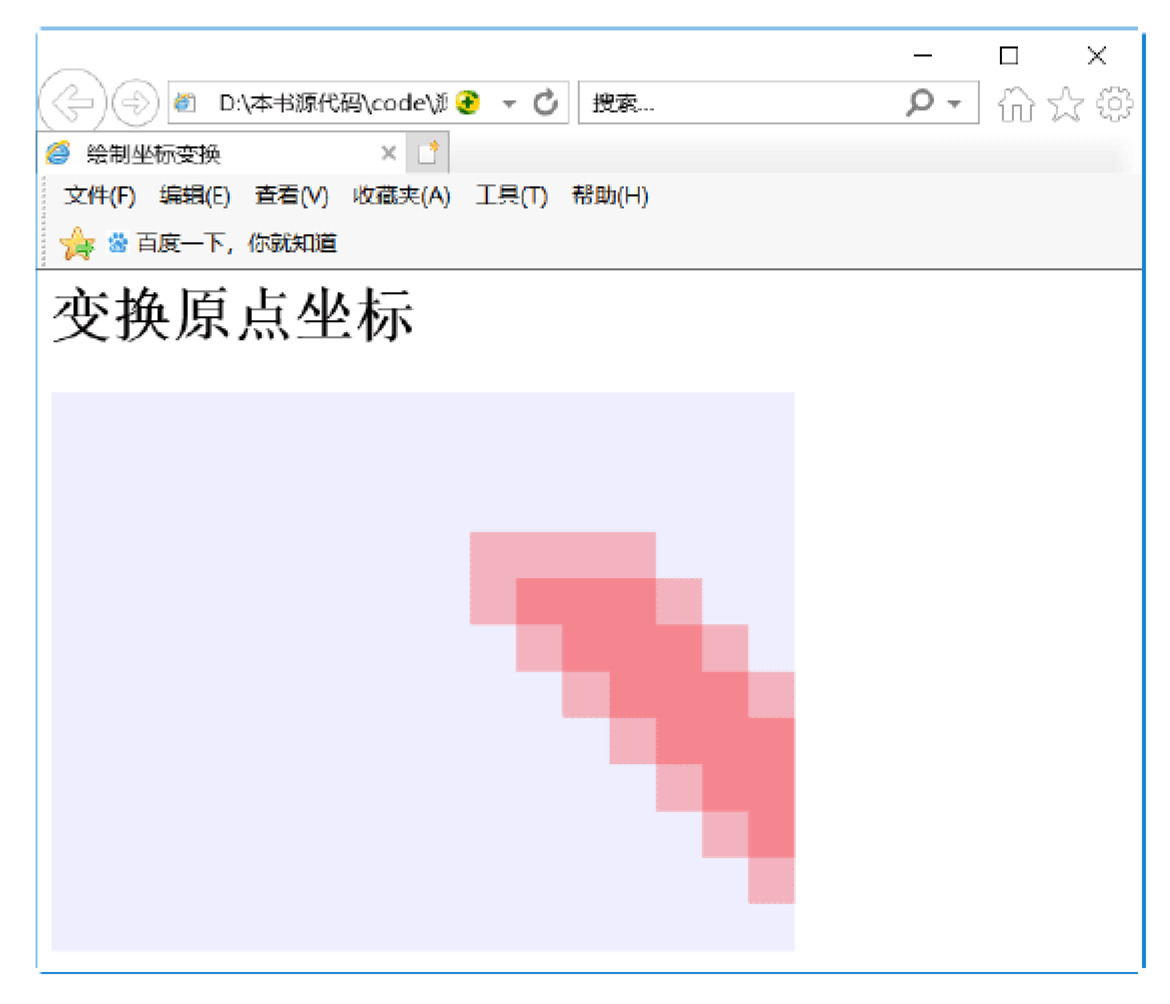


图 7-7 变换坐标原点

7.4.2 图形缩放

对于变形图形来说，其中最常用的方式就是对图形进行缩放，即以原来图形为参考，放大或缩小图形，从而增加效果。

如果要实现图形缩放，就需要使用 scale(x,y)函数，该函数带有两个参数，分别代表在 x,y 两个方向上的值。每个参数在 canvas 显示图像的时候，向其传递在本方向轴上图像要放大（或缩小）的量。如果 x 值为 2，就代表所绘制图像中全部元素都会变成两倍宽。如果 y 值为 0.5，则绘制出来的图像全部元素都会变成之前的一半高。

【例 7.8】（实例文件：ch07\7.8.html）

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图形缩放</title>
<script>
    function draw(id)
    {
        var canvas=document.getElementById(id);
        if(canvas==null)
        return false;
        var context=canvas.getContext('2d');
        context.fillStyle="#eeeeff";
        context.fillRect(0,0,400,300);
```

```
            context.translate(200,50);
            context.fillStyle='rgba(255,0,0,0.25)';
            for(var i=0;i<50;i++){
                context.scale(3,0.5);
                context.fillRect(0,0,100,50);
            }
        }
    </script>
    </head>
    <body onload="draw('canvas');">
    <h1>图形缩放</h1>
    <canvas id="canvas" width="400" height="300" />
    </body>
    </html>
```

上面代码中，实现缩放操作是放在 for 循环中完成的。在此循环中，以原来图形为参考物，使其在 X 轴方向增加为 3 倍宽，y 轴方向上变为原来的一半。

在 IE 11.0 中浏览效果如图 7-8 所示，可以看到在一个指定方向上绘制了多个矩形。

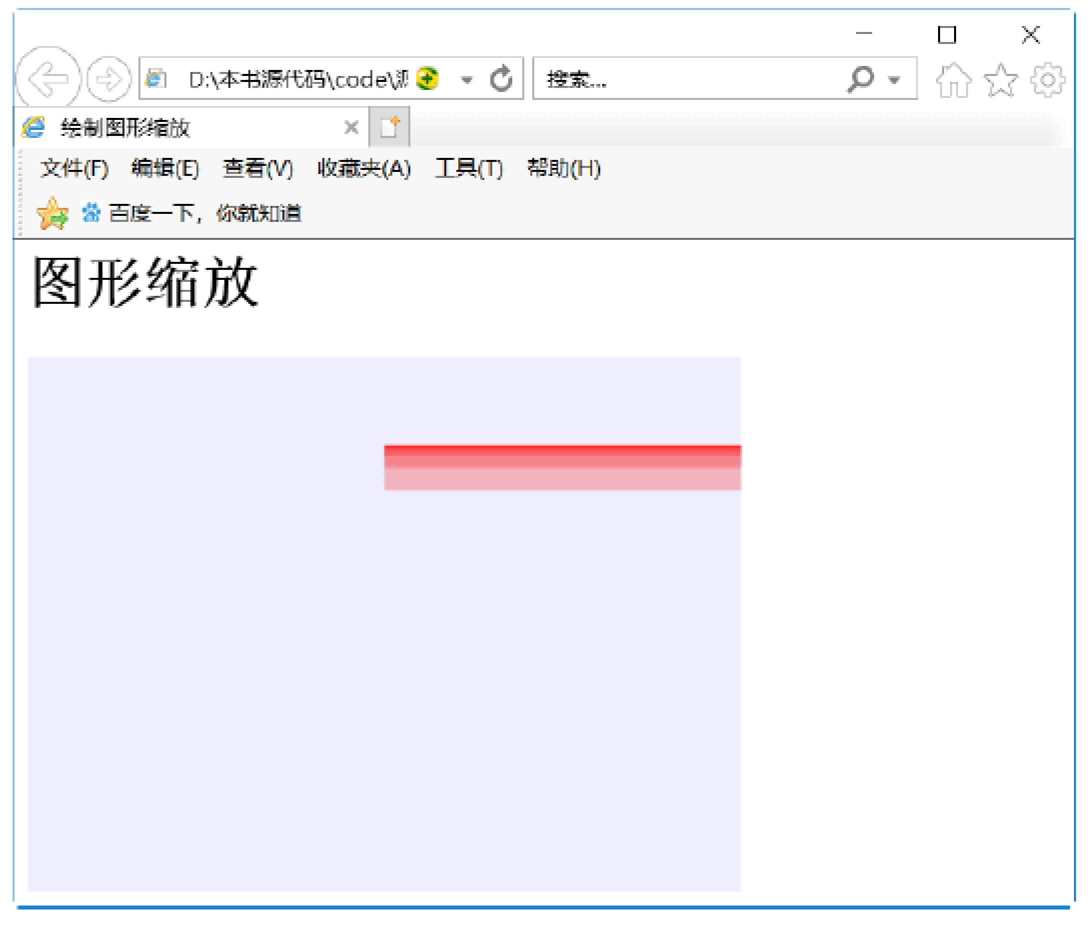


图 7-8　图形缩放

7.4.3　旋转图形

变换操作并不限于缩放和平移，还可以使用函数 context.rotate(angle)来旋转图像，甚至可以直接修改底层变换矩阵以完成一些高级操作，如剪裁图像的绘制路径。

例如，context.rotate(1.57)表示旋转角度参数以弧度为单位。rotate()函数默认地从左上端的（0,0）开始旋转，通过指定一个角度，改变了画布坐标和 Web 浏览器中的<canvas>元素像素之间的映射，使得任意后续绘图在画布中都显示为旋转的，它并没有旋转<canvas>元素本身。注意，这个角度是用弧度指定的。

【例 7.9】（实例文件：ch07\7.9.html）

```
<!DOCTYPE html>
<html>
<head>
<title>绘制旋转图像</title>
<script>
    function draw(id)
    {
        var canvas=document.getElementById(id);
        if(canvas==null)
        return false;
        var context=canvas.getContext('2d');
        context.fillStyle="#eeeeff";
        context.fillRect(0,0,400,300);
        context.translate(200,50);
        context.fillStyle='rgba(255,0,0,0.25)';
        for(var i=0;i<50;i++){
            context.rotate(Math.PI/10);
            context.fillRect(0,0,100,50);
        }
    }
</script>
</head>
<body onload="draw('canvas');">
<h1>旋转图形</h1>
<canvas id="canvas" width="400" height="300" />
</body>
</html>
```

上面代码中，使用 rotate 函数在 for 循环中对多个图形进行了旋转，且旋转角度相同。在 IE 11.0 中浏览效果如图 7-9 所示，可以看到多个矩形以中心弧度为原点进行旋转。

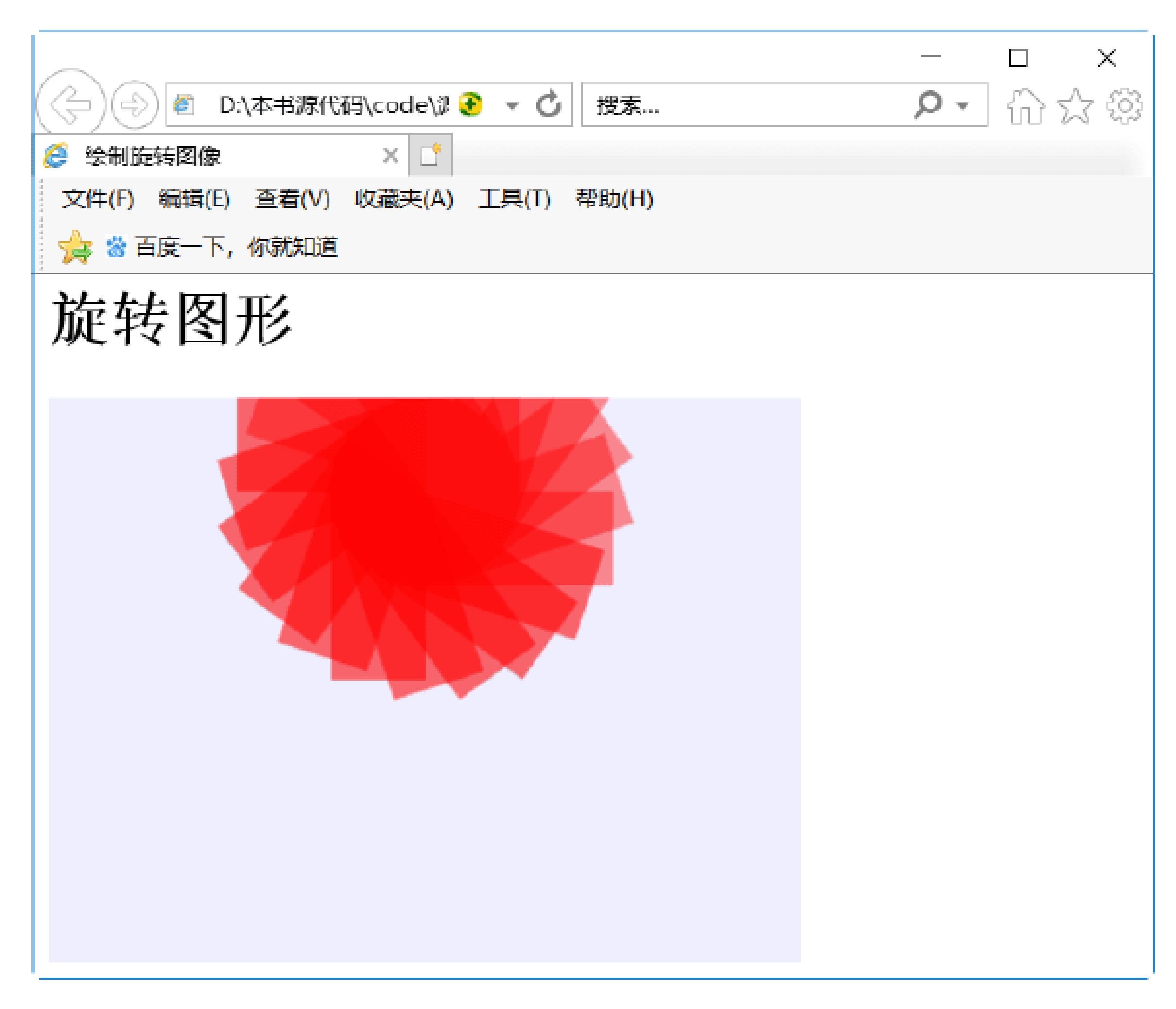


图 7-9　旋转图形

7.5　图形组合

在前面介绍的知识里面可以将一个图形画在另一个之上，大多数情况下是不够的，这样会受制于图形的绘制顺序。不过，我们可以利用 globalCompositeOperation 属性来改变这些做法，不仅可以在已有图形后面再画新图形，还可以用来遮盖，清除（比 clearRect 函数方便得多）某些区域。

其语法格式如下：

```
globalCompositeOperation = type
```

表示设置不同形状的组合类型，其中 type 表示方的图形是已经存在的 canvas 内容，圆的图形是新的形状，其默认值为 source-over，表示在 canvas 内容上面画新的形状。

属性值 type 的含义如表 7-6 所示。

表 7-6　属性值 type 的含义

属性值	说明
source-over(default)	这是默认设置，新图形会覆盖在原有内容之上
destination-over	会在原有内容之下绘制新图形
source-in	新图形会仅仅出现与原有内容重叠的部分，其他区域都变成透明的
destination-in	原有内容中与新图形重叠的部分会被保留，其他区域都变成透明的
source-out	结果是只有新图形中与原有内容不重叠的部分会被绘制出来
destination-out	原有内容中与新图形不重叠的部分会被保留
source-atop	新图形中与原有内容重叠的部分会被绘制，并覆盖于原有内容之上
destination-atop	原有内容中与新内容重叠的部分会被保留，并在原有内容之下绘制新图形
lighter	两图形中重叠部分作加色处理
darker	两图形中重叠部分作减色处理
xor	重叠的部分会变成透明
copy	只有新图形会被保留，其他都被清除掉

【例 7.10】实例文件：ch07\7.10.html）

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图形组合</title>
<script>
function draw(id)
```

```
{
 var canvas=document.getElementById(id);
   if(canvas==null)
  return false;
   var context=canvas.getContext('2d');
   var oprtns=new Array(
    "source-atop",
     "source-in",
    "source-out",
     "source-over",
    "destination-atop",
     "destination-in",
    "destination-out",
    "destination-over",
     "lighter",
    "copy",
    "xor"
   );
  var i=10;
   context.fillStyle="blue";
  context.fillRect(10,10,60,60);
   context.globalCompositeOperation=oprtns[i];
  context.beginPath();
   context.fillStyle="red";
  context.arc(60,60,30,0,Math.PI*2,false);
  context.fill();
}
</script>
</head>
<body onload="draw('canvas');">
<h1>图形组合</h1>
<canvas id="canvas" width="400" height="300" />
</body>
</html>
```

在上面的代码中，首先创建了一个 oprtns 数组，用于存储 type 的 12 个值，然后绘制一个矩形，并使用 content 上下文对象设置了图形的组合方式，即采用新图形显示其他被清除的方式，最后使用 arc 绘制了一个圆。

在 IE 11.0 中浏览效果如图 7-10 所示，在显示页面上绘制了一个矩形和圆，但矩形和圆接触的地方以空白显示。



图 7-10 图形组合

7.6 绘制带阴影的图形

在画布 canvas 上绘制带有阴影效果的图形非常简单，只需要设置几个属性即可。这几个属性分别为 shadowOffsetX、shadowOffsetY、shadowBlur 和 shadowColor，其中属性 shadowColor 表示阴影颜色，其值和 CSS 颜色值一致；shadowBlur 表示设置阴影模糊程度，此值越大，阴影越模糊；shadowOffsetX 和 shadowOffsetY 属性表示阴影的 x 和 y 偏移量，单位是像素。

【例 7.11】实例文件：ch07\7.11.html）

```
<!DOCTYPE html>
<html>
  <head>
  <title>绘制阴影效果图形</title>
 </head>
 <body>
     <canvas id="my canvas" width="200" height="200" style="border:1px solid #ff0000"></canvas>
     <script type="text/javascript">
         var elem = document.getElementById("my canvas");
         if (elem && elem.getContext)  {
             var context = elem.getContext("2d");
             //shadowOffsetX 和 shadowOffsetY：阴影的 x 和 y 偏移量，单位是像素。
             context.shadowOffsetX = 15;
             context.shadowOffsetY = 15;
             //hadowBlur：设置阴影模糊程度。此值越大，阴影越模糊。其效果和 Photoshop 的高斯模糊滤镜相同。
             context.shadowBlur  = 10;
             //shadowColor：阴影颜色。其值和 CSS 颜色值一致。
             //context.shadowColor  = 'rgba(255, 0, 0, 0.5)';  或下面的十六进制的表示方法
             context.shadowColor = '#f00';
```

```
                context.fillStyle  = '#00f';
                context.fillRect(20, 20, 150, 100);
            }
        </script>
    </body>
</html>
```

在 IE 11.0 中浏览效果如图 7-11 所示，在显示页面上显示了一个蓝色矩形，其阴影为红色矩形。

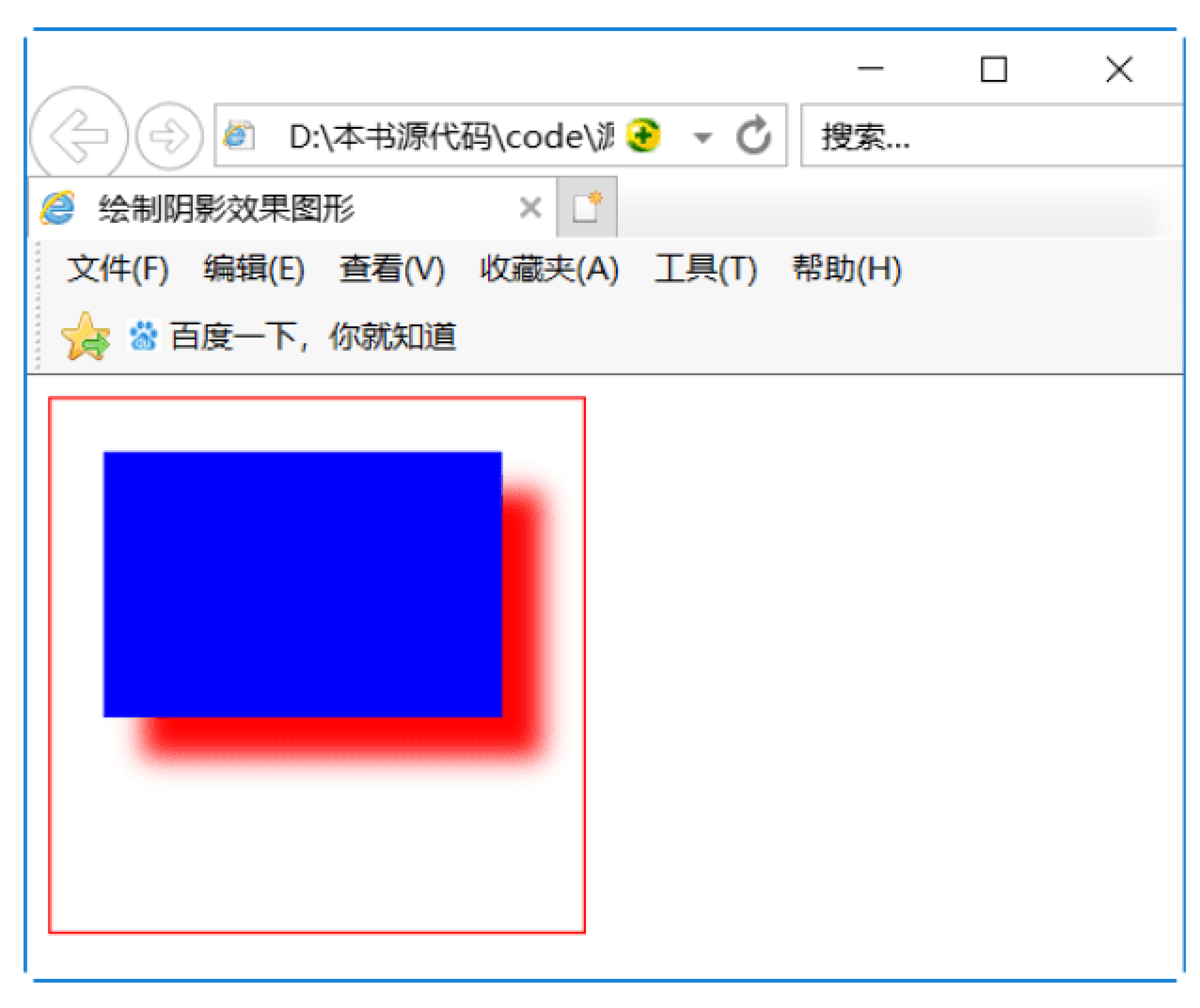


图 7-11　带有阴影的图形

7.7　使用图像

画布 canvas 有一项功能就是可以引入图像，它可以用于图片合成或制作背景等，而目前仅可以在图像中加入文字。只要是 Geck 支持的图像（如 PNG、GIF、JPEG 等）都可以引入到 canvas 中，而且其他的 canvas 元素也可以作为图像的来源。

7.7.1　绘制图像

要在画布 canvas 上绘制图像，需要先有一张图片。这张图片可以是已经存在的<img>元素，或者通过 JS 创建，无论采用哪种方式，都需要在绘制 canvas 之前完全加载这张图片。浏览器通常会在页面脚本执行的同时异步加载图片，如果试图在图片未完全加载之前就将其呈现到 canvas 上，那么 canvas 将不会显示任何图片。

捕获和绘制图形完全是通过 drawImage 函数完成的，它可以接受不同的 HTML 参数，具体含义如表 7-7 所示。

表 7-7 drawImage 函数

函数	说明
drawIamge(image,dx,dy)	接受一张图片，并将其画到 canvas 中。给出的坐标（dx,dy）代表图片的左上角，如坐标（0，0）将把图片画到 canvas 的左上角
drawIamge(image,dx,dy,dw,dh)	接受一张图片，将其缩放为宽度 dw 和高度 dh，然后把它画到 canvas 上的(dx,dy)位置
drawIamge(image,sx,sy,sw,sh, dx,dy,dw,dh)	接受一张图片，通过参数（sx,sy,sw,sh）指定图片裁剪的范围，缩放到(dw,dh)的大小，最后把它画到 canvas 上的(dx,dy)位置

【例 7.12】实例文件：ch07\7.12.html）

```
<!DOCTYPE html>
<html>
<head><title>绘制图像</title></head>
<body>
<canvas id="canvas" width="300" height="200" style="border:1px solid blue">
Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
window.onload=function(){ //使用窗口的onload加载事件
  var ctx=document.getElementById("canvas").getContext("2d"); //创建上下文
对象
  var img=new Image();     //创建Image对象img
  img.src="01.jpg";        //使用img对象的属性src设置图片来源
  img.onload=function(){
    ctx.drawImage(img,0,0); //使用drawImage画出当前的图像
  }
}
</script>
</body>
</html>
```

在上面代码中，使用窗口的 onload 加载事件，即页面被加载时执行函数。在函数中创建上下文对象 ctx 并创建 Image 对象 img，然后使用 img 对象的属性 src 设置图片来源，最后使用 drawImage 画出当前的图像。

在 IE 11.0 中浏览效果如图 7-12 所示，在显示页面上绘制了一个图像并在画布中显示。

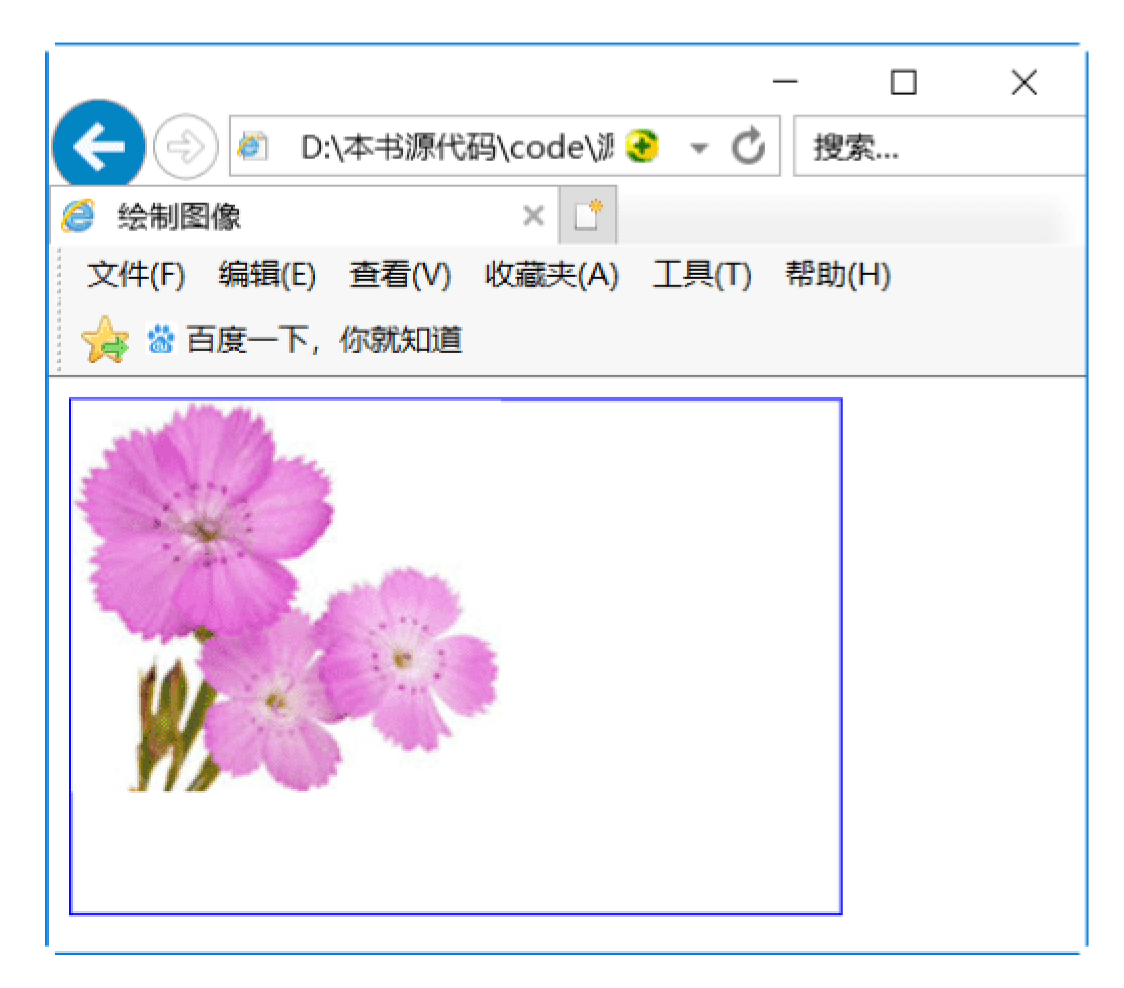


图 7-12　绘制图像

7.7.2　图像平铺

使用画布 canvas 绘制图像有很多用处，其中一个用处就是将绘制的图像作为背景图片使用。在做背景图片时，如果显示图片的区域大小不能直接设置，通常将图片以平铺的方式显示。

HTML5 Canvas API 支持图片平铺，此时需要调用 createPattern 函数，即调用 createPattern 函数来替代之前的 drawImage 函数。

函数 createPattern 的语法格式如下：

```
createPattern(image,type)
```

其中 image 表示要绘制的图像，type 表示平铺的类型。平铺类型如表 7-8 所示。

表 7-8　平铺类型

参数值	说明
no-repeat	不平铺
repeat-x	横方向平铺
repeat-y	纵方向平铺
repeat	全方向平铺

【例 7.13】实例文件：ch07\7.13.html）

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图像平铺</title>
</head>
<body onload="draw('canvas');">
```

```
<h1>图形平铺</h1>
<canvas id="canvas" width="400" height="300"></canvas>
<script>
    function draw(id){
        var canvas=document.getElementById(id);
        if(canvas==null){
            return false;
        }
        var context=canvas.getContext('2d');
        context.fillStyle="#eeeeff";
        context.fillRect(0,0,400,300);
        image=new Image();
        image.src="01.jpg";
        image.onload=function(){
            var ptrn=context.createPattern(image,'repeat');
            context.fillStyle=ptrn;
            context.fillRect(0,0,400,300);
        }
    }
</script>
</body>
</html>
```

上面代码中，使用 fillRect 创建了一个宽度为 400，高度为 300，左上角坐标位置为（0，0）的矩形，然后创建了一个 Image 对象。src 表示连接一个图像源，使用 createPattern 绘制一个图像，其方式是以完全平铺，并将这个图像作为一个模式填充到矩形中，最后绘制这个矩形，此矩形大小完全覆盖原来的图形。

在 IE 11.0 中浏览效果如图 7-13 所示，在显示页面上绘制了一个图像，其图像以平铺的方式添满整个矩形。



图 7-13　图像平铺

7.7.3 图像裁剪

在处理图像时经常会遇到裁剪这种需求，即在画布上裁剪出一块区域，该区域是在裁剪动作 clip 之前，由绘图路径设置的，可以是方形、圆形、五星形和其他任何可以绘制的轮廓形状。所以，裁剪路径其实就是绘图路径，只不过这个路径不是拿来绘图的，而是设置显示区域和遮挡区域的一个分界线。

完成对图像的裁剪，可能要用到 clip 函数。clip 函数表示给 canvas 设置一个剪辑区域，在调用 clip 函数之后的代码只对这个设置的剪辑区域有效，不会影响其他地方，这个函数在进行局部更新时很有用。默认情况下，剪辑区域是一个左上角在（0, 0），宽和高分别等于 canvas 元素的宽和高的矩形。

【例 7.14】实例文件：ch07\7.14.html）

```
<!DOCTYPE html>
< html>
<head>
<title>绘制图像裁剪</title>
<script type="text/javascript" src="script.js"></script>
</head>
<body onload="draw('canvas');">
<h1>图像裁剪实例</h1>
<canvas id="canvas" width="400" height="300"></canvas>
<script>
    function draw(id){
        var canvas=document.getElementById(id);
        if(canvas==null){
            return false;
        }
        var context=canvas.getContext('2d');
        var gr=context.createLinearGradient(0,400,300,0);
        gr.addColorStop(0,'rgb(255,255,0)');
        gr.addColorStop(1,'rgb(0,255,255)');
        context.fillStyle=gr;
        context.fillRect(0,0,400,300);
        image=new Image();
        image.onload=function(){
            drawImg(context,image);
        };
        image.src="01.jpg";
    }
    function drawImg(context,image){
        create8StarClip(context);
        context.drawImage(image,-50,-150,300,300);
    }
    function create8StarClip(context){
```

```
            var n=0;
            var dx=100;
            var dy=0;
            var s=150;
            context.beginPath();
            context.translate(100,150);
            var x=Math.sin(0);
            var y=Math.cos(0);
            var dig=Math.PI/5*4;
            for(var i=0;i<8;i++){
                var x=Math.sin(i*dig);
                var y=Math.cos(i*dig);
                context.lineTo(dx+x*s,dy+y*s);
            }
            context.clip();
        }
    </script>
    </body>
    </html>
```

上面代码中，创建了 3 个 JavaScript 函数，其中 create8StarClip 函数完成了多边的图形创建，其中以此图形作为裁剪的依据；drawImg 函数表示绘制一个图形，其图形带有裁剪区域；draw 函数完成对画布对象的获取，并定义一个线性渐变，然后创建一个 Image 对象。

在 IE 11.0 中浏览效果如图 7-14 所示，在显示页面上绘制一个五边形，图像作为五边形的背景显示，从而实现对象图像的裁剪。

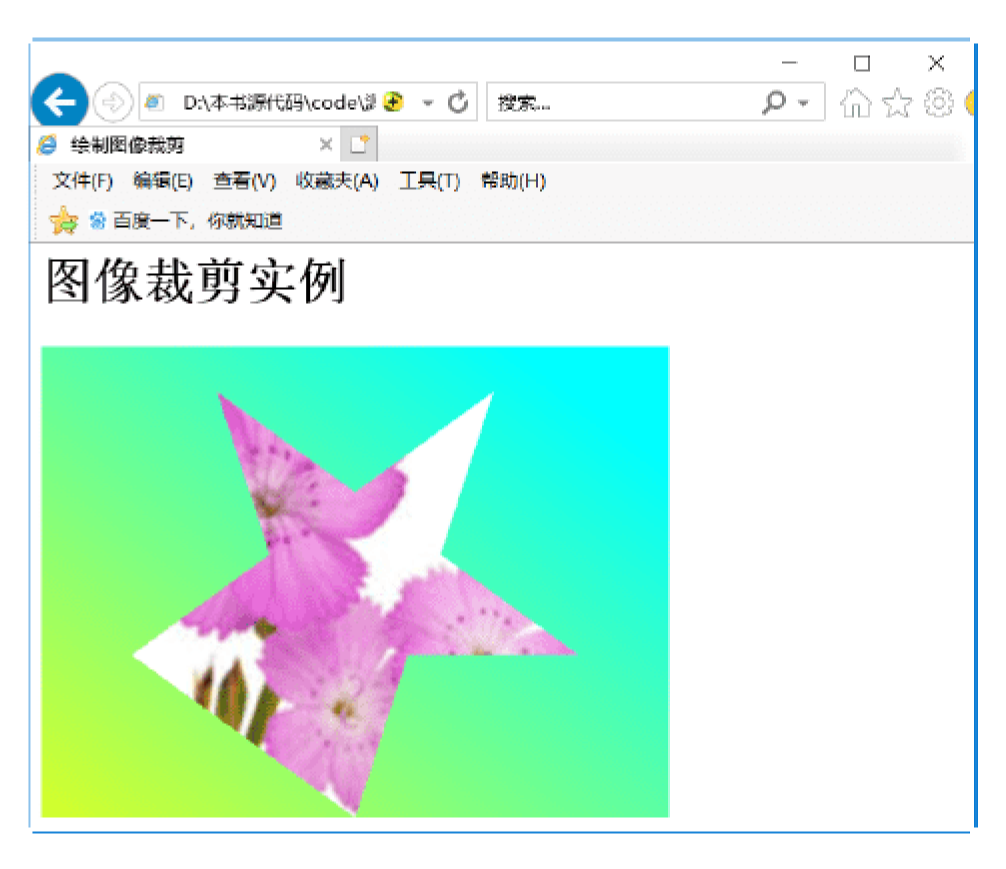


图 7-14　图像裁剪

7.7.4　像素处理

在电脑屏幕上可以看到色彩斑斓的图像，其实这些图像都是由一个个像素点组成的。一个像素对应着内存中的一组连续的二进制位，由于是二进制位，因此每个位上的取值只能是 0 或

1。这样，这组连续的二进制位就可以由 0 和 1 排列组合出很多种情况，而每一种排列组合就决定了这个像素的一种颜色。通常，每个像素点由 4 个字节组成。

这 4 个字节代表的含义分别是，第一个字节决定像素的红色值；第二个字节决定像素的绿色值；第 3 个字节决定像素的蓝色值；第 4 个字节决定像素的透明度值。

在画布中，可以使用 ImageData 对象用来保存图像像素值，它有 width、height 和 data 3 个属性，其中 data 属性就是一个连续数组，图像的所有像素值其实是保存在 data 里面的。

data 属性保存像素值的方法如下：

```
imageData.data[index*4 +0]
imageData.data[index*4 +1]
imageData.data[index*4 +2]
imageData.data[index*4 +3]
```

上面取出了 data 数组中连续相邻的 4 个值，分别代表了图像中第 index+1 个像素的红色、绿色、蓝色和透明度值的大小。需要注意的是，index 从 0 开始，图像中总共有 width×height 个像素，数组中总共保存了 width×height×4 个数值。

画布对象有 3 个函数用来创建、读取和设置 ImageData 对象，如表 7-9 所示。

表 7-9　画布对象函数

函数	说明
createImageData(width, height)	在内存中创建一个指定大小的 ImageData 对象（即像素数组），对象中的像素点都是黑色透明的，即 rgba(0,0,0,0)
getImageData(x, y, width, height)	返回一个 ImageData 对象，这个 IamgeData 对象中包含了指定区域的像素数组
putImageData(data, x, y)	将 ImageData 对象绘制到屏幕的指定区域上

【例 7.15】实例文件：ch07\7.15.html）

```
<!DOCTYPE html>
<html>
<head>
<title>图像像素处理</title>
<script type="text/javascript" src="script.js"></script>
</head>
<body onload="draw('canvas');">
<h1>像素处理示例</h1>
<canvas id="canvas" width="400" height="300"></canvas>
<script>
    function draw(id){
        var canvas=document.getElementById(id);
```

```
            if(canvas==null){
                return false;
            }
            var context=canvas.getContext('2d');
            image=new Image();
            image.src="01.jpg";
            image.onload=function(){
                context.drawImage(image,0,0);
                var
imagedata=context.getImageData(0,0,image.width,image.height);
                for(var i=0,n=imagedata.data.length;i<n;i+=4){
                    imagedata.data[i+0]=255-imagedata.data[i+0];
                    imagedata.data[i+1]=255-imagedata.data[i+2];
                    imagedata.data[i+2]=255-imagedata.data[i+1];

                }
                context.putImageData(imagedata,0,0);
            };
        }
    </script>
    </body>
    </html>
```

在上面代码中，使用 getImageData 函数获取一个 ImageData 对象并包含相关的像素数组，在 for 循环中对像素值重新赋值，最后使用 putImageData 将处理过的图像在画布上绘制出来。

在 IE 11.0 中浏览效果如图 7-15 所示，在页面上显示了一个图像，其图像明显经过像素处理，显示没有原来清晰。

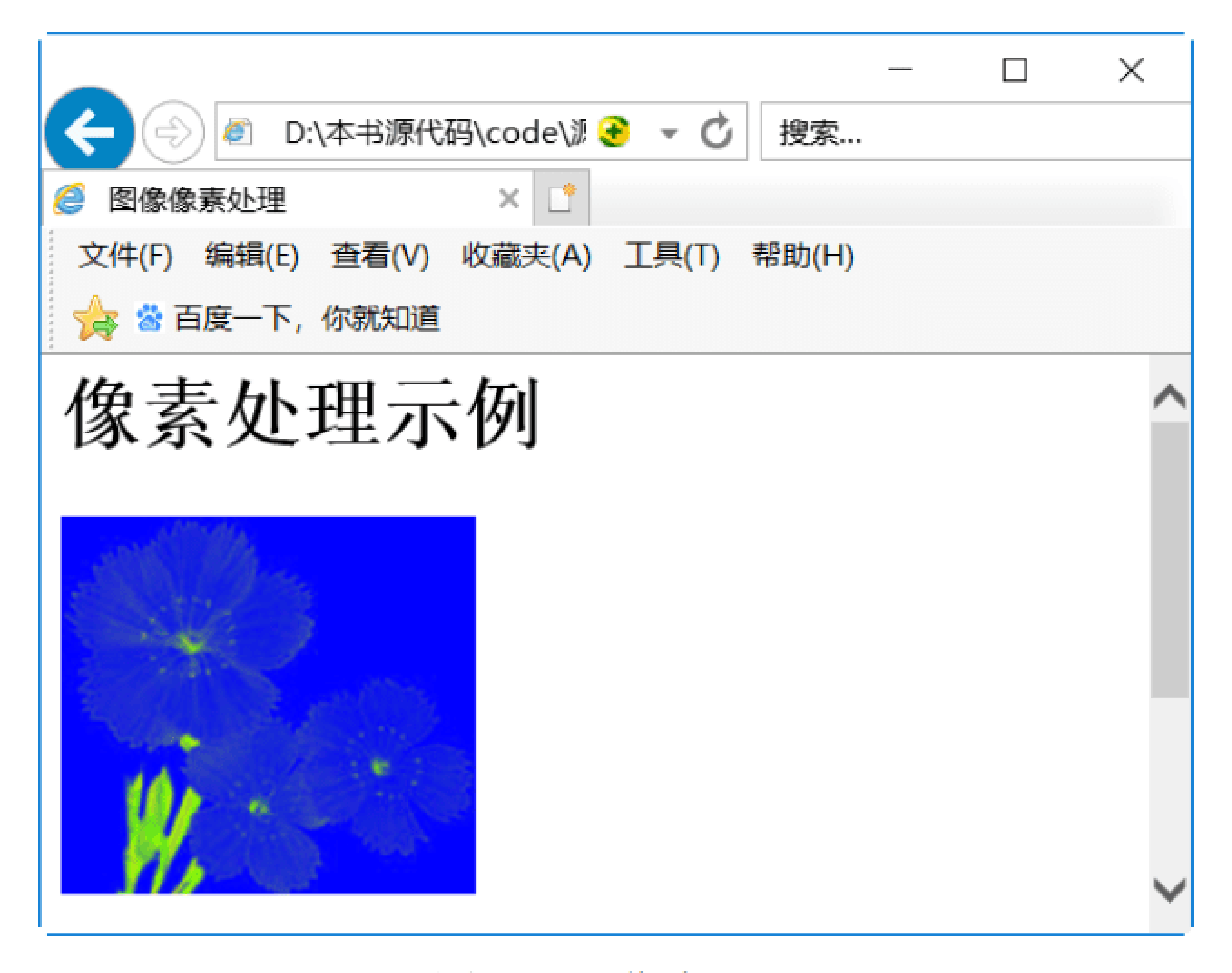


图 7-15　像素处理

7.8 绘制文字

在画布中绘制字符串（文字）的方式，与操作其他路径对象的方式相同，可以描绘文本轮廓和填充文本内部，同时所有能够应用于其他图形的变换和样式都能用于文本。

文本绘制功能由两个函数组成，如表 7-10 所示。

表 7-10 文本绘制函数

函数	说明
fillText(text,x,y,maxwidth)	绘制带 fillStyle 填充的文字、文本参数以及用于指定文本位置的坐标参数。Maxwidth 是可选参数，用于限制字体大小，它会将文本字体强制收缩到指定尺寸
trokeText(text,x,y,maxwidth)	绘制只有 strokeStyle 边框的文字，其参数含义与上一个方法相同
measureText	该函数会返回一个度量对象，其包含了在当前 context 环境下指定文本的实际显示宽度

为了保证文本在各浏览器下都能正常显示，在绘制上下文里有以下字体属性。

font 可以是 CSS 字体规则中的任何值，包括字体样式、字体变种、字体大小与粗细、行高和字体名称。

textAlign 控制文本的对齐方式。它类似于（但不完全相同）CSS 中的 text-align。可能的取值为 start、end、left、right、和 center。

textBaseline 控制文本相对于起点的位置。可能的取值有 top、hanging、middle、alphabetic、ideographic 和 bottom。对于简单的英文字母，可以放心使用 top、middle 或 bottom 作为其文本基线。

【例 7.16】实例文件：ch07\7.7.html）

```
    <!DOCTYPE html>
    <html>
      <head>
       <title>Canvas</title>
      </head>
      <body>
        <canvas id="my_canvas" width="200" height="200" style="border:1px
solid #ff0000"></canvas>
        <script type="text/javascript">
            var elem = document.getElementById("my canvas");
            if (elem && elem.getContext)  {
                var context = elem.getContext("2d");
                context.fillStyle  = '#00f';
```

```
                    //font：文字字体，同 CSSfont-family 属性
                    context.font = 'italic  30px 微软雅黑';  //斜体 30像素 微软雅黑字
体
                    //textAlign：文字水平对齐方式。可取属性值：start, end, left,right,
center。默认值:start.
                    context.textAlign = 'left';
                    //文字竖直对齐方式。可取属性值：top, hanging, middle,alphabetic,
ideographic, bottom。默认值：alphabetic
                    context.textBaseline = 'top';
                    //要输出的文字内容，文字位置坐标，第4个参数为可选选项——最大宽度。如果需
要的话，则浏览器会缩减文字以让它适应指定宽度
                    context.fillText  ('祖国生日快乐!', 0, 0,50);  //有填充
                    context.font     = 'bold 30px sans-serif';
                    context.strokeText('祖国生日快乐!', 0, 50,100);  //只有文字边框
               }
          </script>
      </body>
    </html>
```

在 IE 11.0 中浏览效果如图 7-16 所示，在页面上显示了一个画布边框，其中有两个不同的字符串：第一个字符串以斜体显示，其颜色为蓝色；第二个字符串字体颜色为黑色，加粗显示。

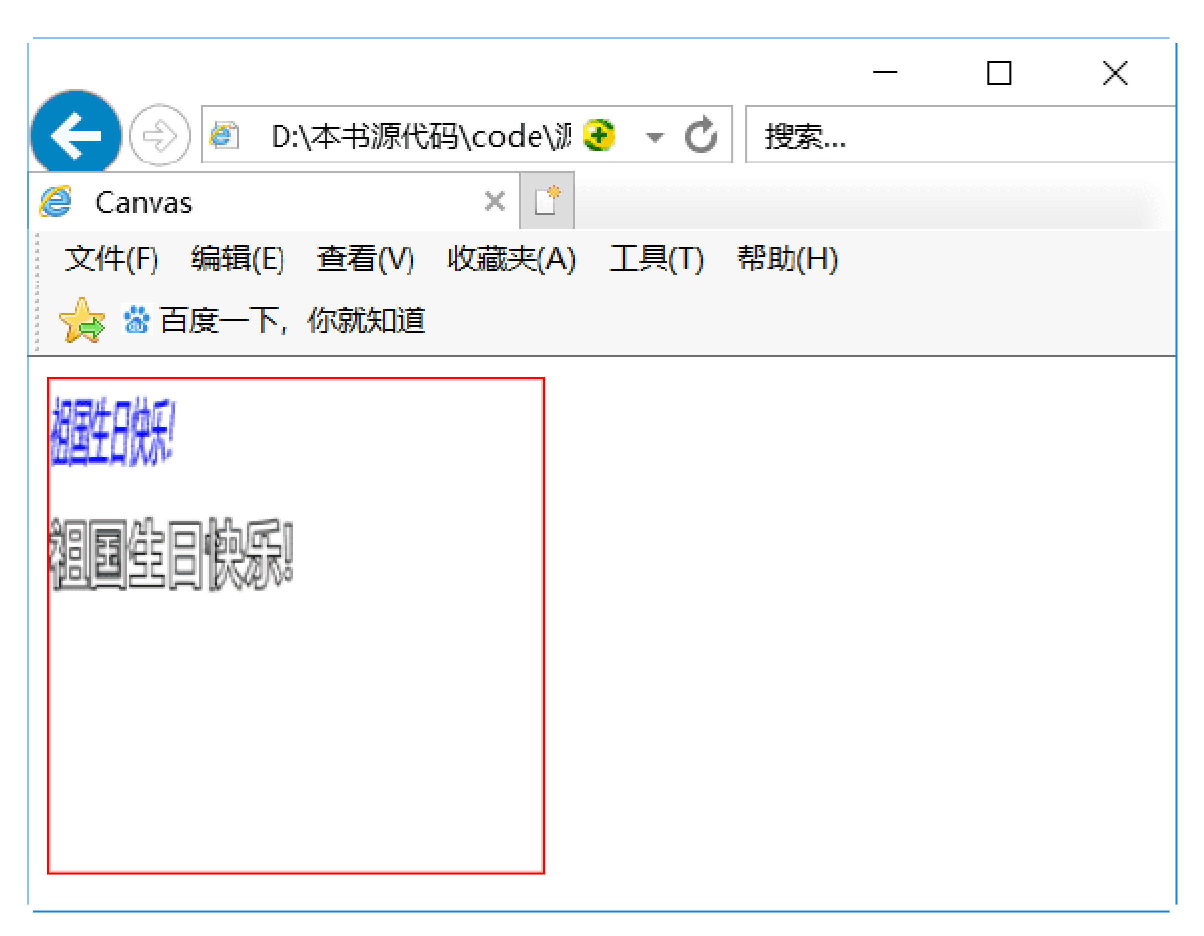


图 7-16　绘制文字

7.9　图形的保存与恢复

在画布对象绘制图形或图像时，可以对这些图形或图形的状态进行改变，即永久保存图形或图像。

7.9.1　保存与恢复状态

在画布对象中，由两个函数管理绘制状态的当前栈，save 函数把当前状态压入栈中，而

restore 从栈顶弹出状态。绘制状态不会覆盖对画布所做的每件事情，其中 save 函数用来保存 canvas 的状态。save 之后，可以调用 Canvas 的平移、放缩、旋转、裁剪等操作。Restore 函数用来恢复 Canvas 之前保存的状态。save 和 restore 要配对使用（restore 可以比 save 少，但不能多），如果 restore 调用次数比 save 多，则会引发 Error。

【例 7.17】实例文件：ch07\7.17.html）

```
<!DOCTYPE html>
<html>
<head><title>保存与恢复</title></head>
<body>
<canvas id="myCanvas" width="500" height="400" style="border:1px solid 
blue">
Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.fillStyle = "rgb(0,0,255)";
ctx.save();
ctx.fillRect(50,50,100,100);
ctx.fillStyle = "rgb(255,0,0)";
ctx.save();
ctx.fillRect(200,50,100,100);
ctx.restore()
ctx.fillRect(350,50,100,100);
ctx.restore();
ctx.fillRect(50, 200, 100, 100);
</script>
</body>
</html>
```

在上面代码中，绘制了 4 个矩形，在第一个矩形绘制之前，定义当前矩形的显示颜色，并将此样式加入到栈中，然后创建了一个矩形。第二个矩形绘制之前，重新定义了矩形显示颜色，并使用 save 将此样式压入到栈中，然后创建了一个矩形。在第三个矩形绘制之前，使用 restore 恢复当前显示颜色，即调用栈中的最上层颜色，绘制矩形。在第 4 个矩形绘制之前，继续使用 restore 函数，调用最后一个栈中元素定义矩形颜色。

在 IE 11.0 中浏览效果如图 7-17 所示，在显示页面上绘制了 4 个矩形，第一个和第 4 个矩形显示为蓝色，第二个和第三个矩形显示为红色。

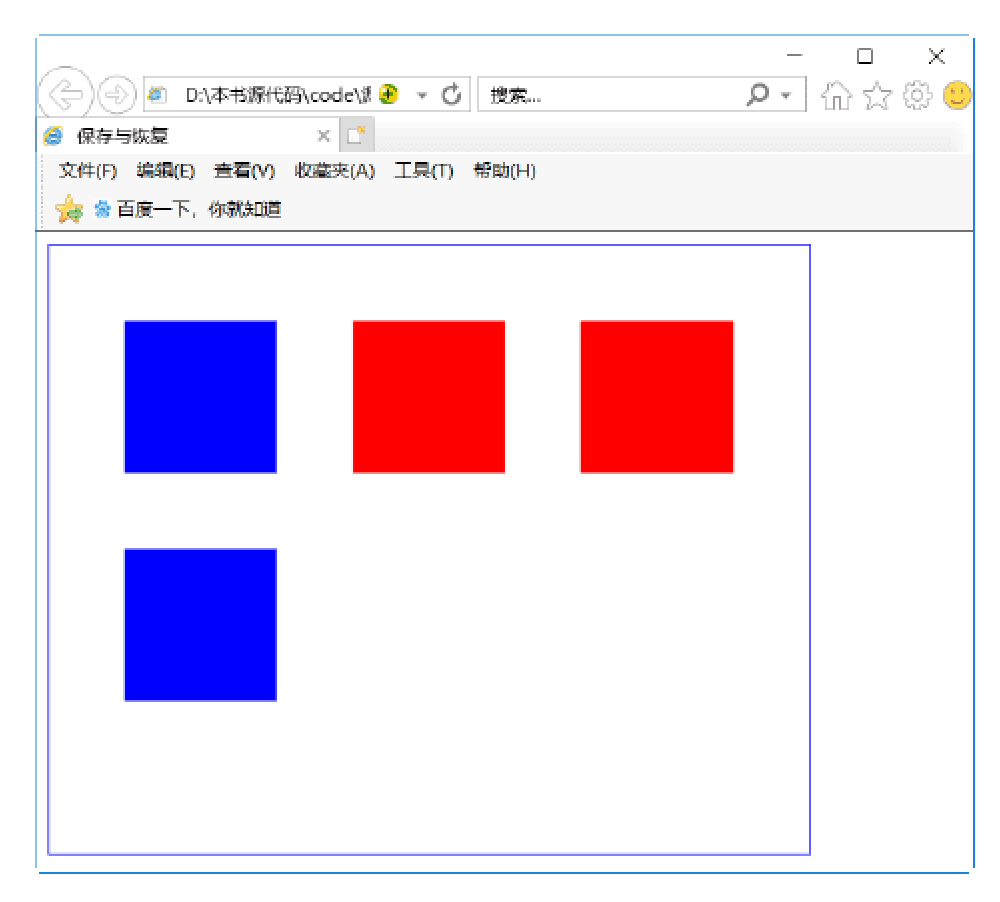


图 7-17 恢复和保存

7.9.2 保存文件

当绘制出漂亮的图形时需要保存这些劳动成果，这时可以将当前的画布元素（而不是 2D 环境）的当前状态导出到数据 URL。导出很简单，可以利用 toDataURL 函数完成，它可以以不同的图片格式来调用。目前 png 格式是规范定义的格式，通常浏览器也支持其他的格式。

目前 Firefox 和 Opera 浏览器只支持 png 格式，Safari 支持 gif、png 和 jpg 格式。大多数浏览器支持读取 base64 编码内容。URL 的格式如下：

```
data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAfQAAAH0CAYAAADL1t
```

它先以一个 data 开始，然后是 mine 类型，之后是编码和 base64，最后是原始数据。这些原始数据就是画布元素所要导出的内容，并且浏览器能够将数据编码为真正的资源。

【例 7.18】实例文件：ch07\7.18.html）

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="500" height="500" style="border:1px solid 
blue">
Your browser does not support the canvas element</canvas>
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.fillStyle='rgb(0,0,255)';
cxt.fillRect(0,0,cxt.canvas.width,cxt.canvas.height);
cxt.fillStyle="rgb(0,255,0)";
cxt.fillRect(10,20,50,50);
window.location=cxt.canvas.toDataURL(image/png');
</script>
```

```
</body>
</html>
```

在上面代码中，使用 canvas.toDataURL 语句将当前绘制图像保存到 URL 数据中。在 Firefox 62.0 中浏览效果如图 7-18 所示，在显示页面中无任何数据显示，并且提示无法显示该页面。此时需要注意的是鼠标指向的位置，即地址栏中 URL 数据。

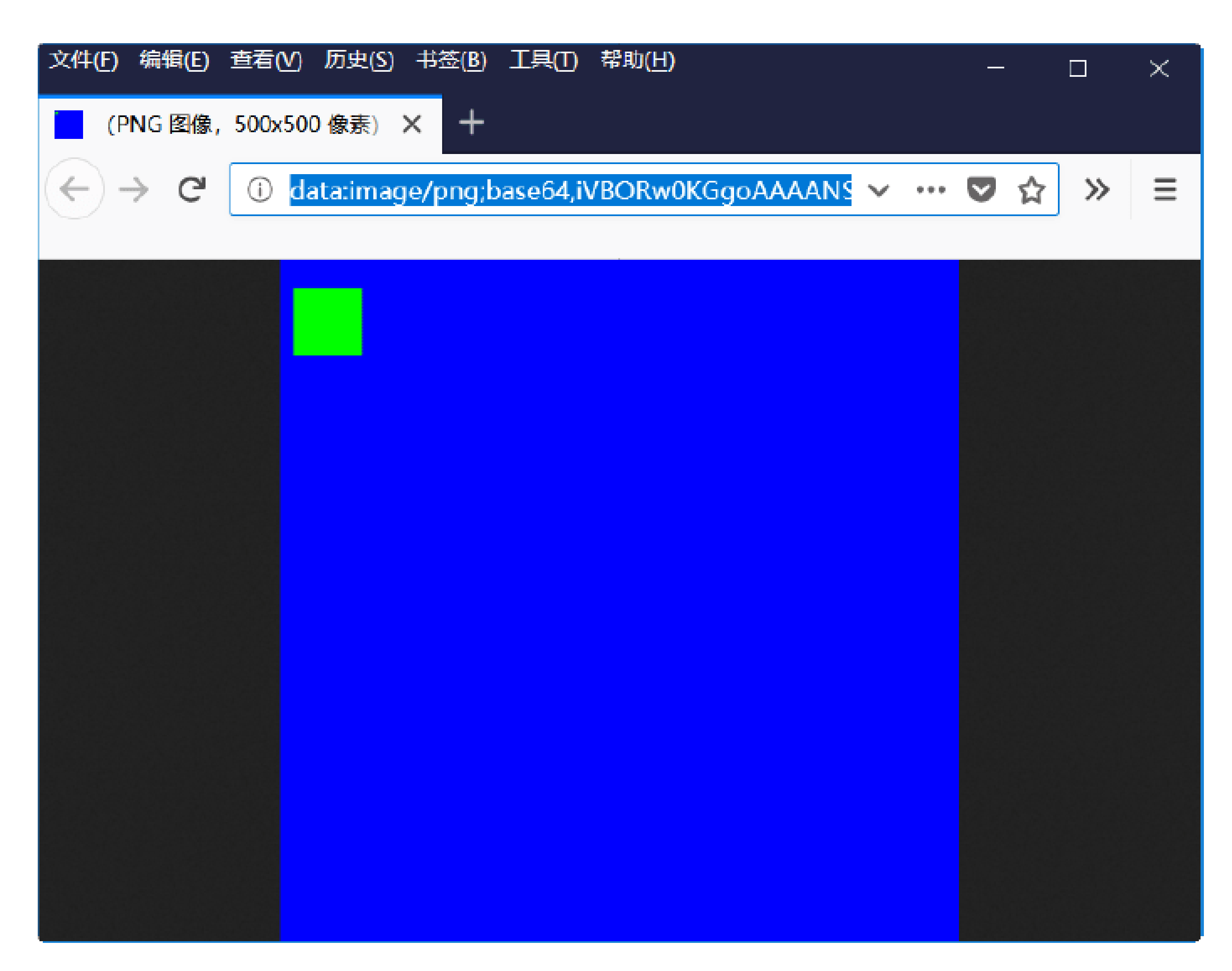


图 7-18　保存图形

7.10　综合实例 1——绘制商标

绘制商标是 canvas 画布的用途之一，可以绘制 adidas 和 nike 商标。nike 的图标比 adidas 的图标复杂得多，adidas 都是直线组成，而 nike 则多了曲线。实现本实例的步骤如下：

01 分析需求。

要绘制两条曲线，需要找到曲线的参考点（参考点决定了曲线的曲率），即慢慢移动，然后看效果，反反复复。quadraticCurveTo(30,79,99,78)函数有两组坐标：第一组坐标为控制点，决定曲线的曲率；第二组坐标为终点。

02 构建 HTML，实现 canvas 画布。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制商标</title>
</head>
```

```
    <body>
    <canvas id="adidas" width="375px" height="132px" style="border:1px solid
#000;"></canvas>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 7-19 所示，此时只显示一个画布边框，其内容还没有绘制。

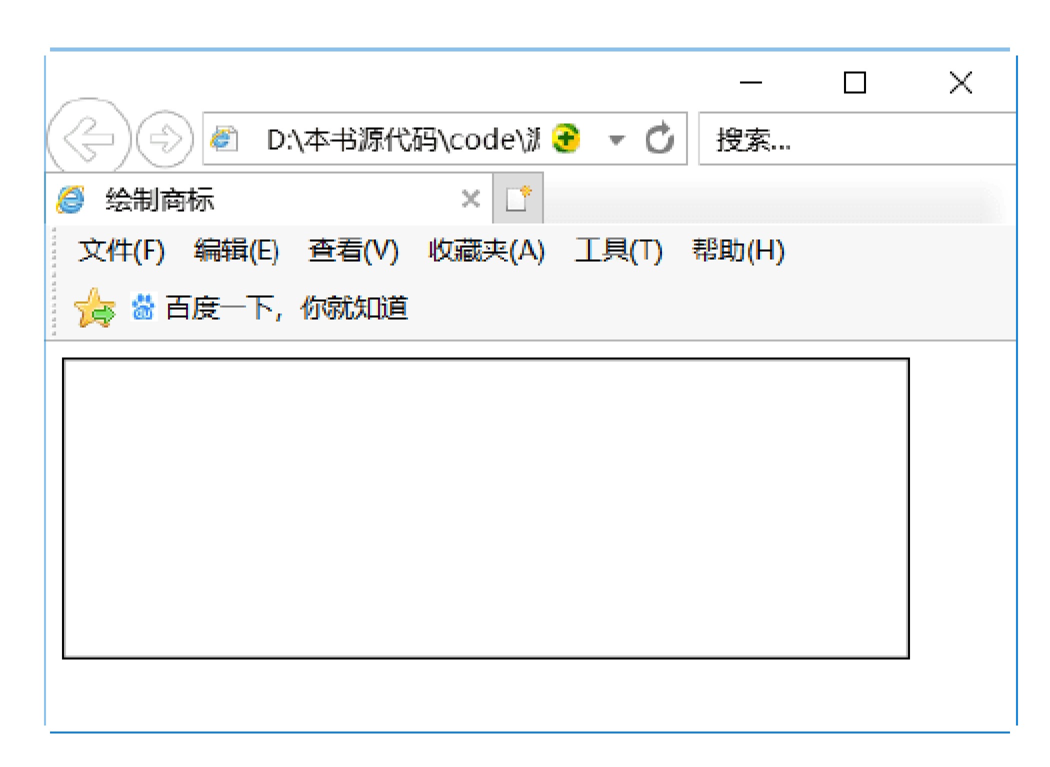


图 7-19　定义画布边框

03 JS 实现基本图形。

```
    <script>
    function drawAdidas(){
        //取得convas元素及其绘图上下文
        var canvas=document.getElementById('adidas');
        var context=canvas.getContext('2d');
        //保存当前绘图状态
        context.save();
        //开始绘制打勾的轮廓
        context.beginPath();
        context.moveTo(53,0);
        //绘制上半部分曲线，第一组坐标为控制点，决定曲线的曲率；第二组坐标为终点
        context.quadraticCurveTo(30,79,99,78);
        context.lineTo(371,2);
        context.lineTo(74,134);
        context.quadraticCurveTo(-55,124,53,0);
        //用红色填充
        context.fillStyle="#da251c";
        context.fill();
        //用3像素深红线条描边
        context.lineWidth=3;
        //连接处平滑
        context.lineJoin='round';
        context.strokeStyle="#d40000";
        context.stroke();
        //恢复原有绘图状态
        context.restore();
    }
```

```
window.addEventListener("load",drawAdidas,true);
</script>
```

在 IE 11.0 中浏览效果如图 7-20 所示，显示了一个商标图案，颜色为红色。

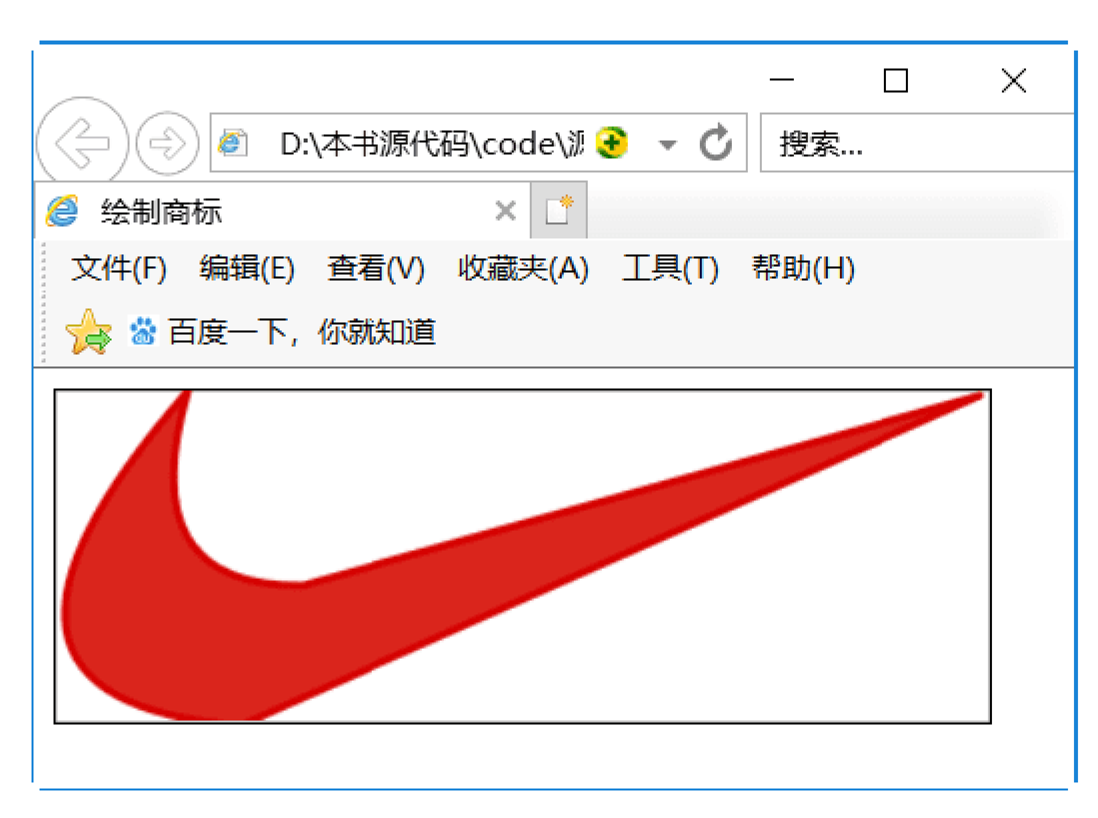


图 7-20　绘制商标

7.11　综合实例 2——绘制火柴棒人物

漫画中比较常见的一种图形就是火柴棒人，通过简单的几个笔画，就可以绘制一个传神的动漫人物。使用 canvas 和 JavaScript 同样可以绘制一个火柴棒人物。具体步骤如下：

01 分析需求

一个火柴棒人，由脸部和身躯组成。脸部是一个圆形，包括眼睛和嘴；身躯是几条直线，包括手和腿等。实际上此案例就是绘制圆形、弧度和直线的组合。

02 实现 HTML 页面，定义画布 canvas

```
<!DOCTYPE html>
<html>
<title>绘制火柴棒人</title>
<body>
<canvas id="myCanvas" width="500" height="300" style="border:1px solid
blue">
Your browser does not support the canvas element.
</canvas>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 7-21 所示，页面显示了一个画布边框。



图 7-21　定义画布边框

03 实现头部轮廓绘制

```
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.beginPath();
cxt.arc(100,50,30,0,Math.PI*2,true);
cxt.fill();
</script>
```

这会产生一个实心的、填充的头部，即圆形。在 arc 函数中，x 和 y 的坐标为（100，50），半径为 30 像素，另外两个参数的弧度为弧度的开始和结束，第 6 个参数表示绘制弧形的方向，即顺时针和逆时针方向。

在 IE 11.0 中浏览效果如图 7-22 所示，页面显示了一个实心圆，其颜色为黑色。

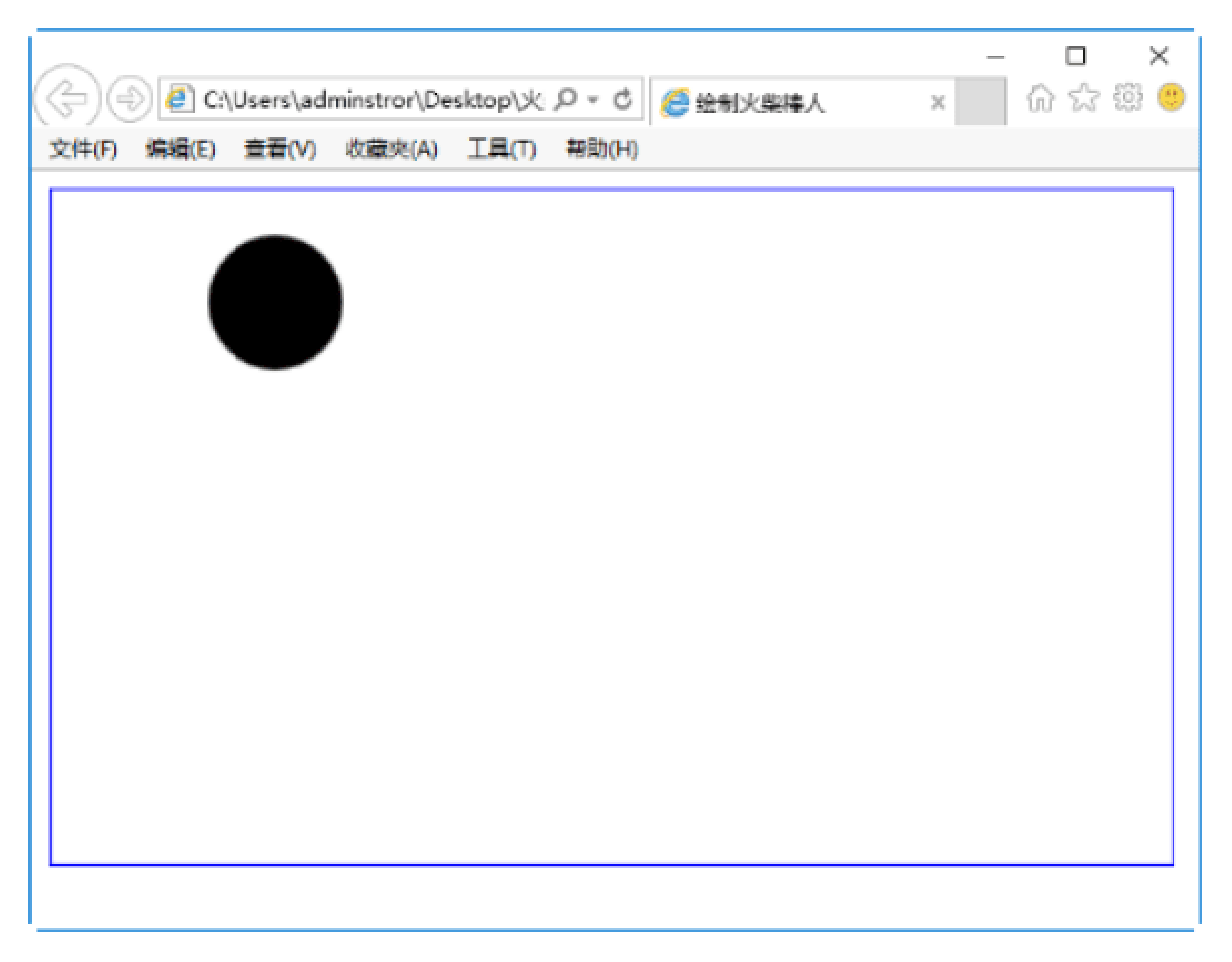


图 7-22　绘制头部轮廓

04 JS 绘制笑脸。

```
cxt.beginPath();
cxt.strokeStyle='#c00';
cxt.lineWidth=3;
cxt.arc(100,50,20,0,Math.PI,false);
cxt.stroke();
```

此处使用 beginPath 方法，表示重新绘制，并设置线条宽度，然后绘制一个弧形，这个弧形是从嘴部开始的。

在 IE 11.0 中浏览效果如图 7-23 所示，页面上显示了一个半圆式笑脸。

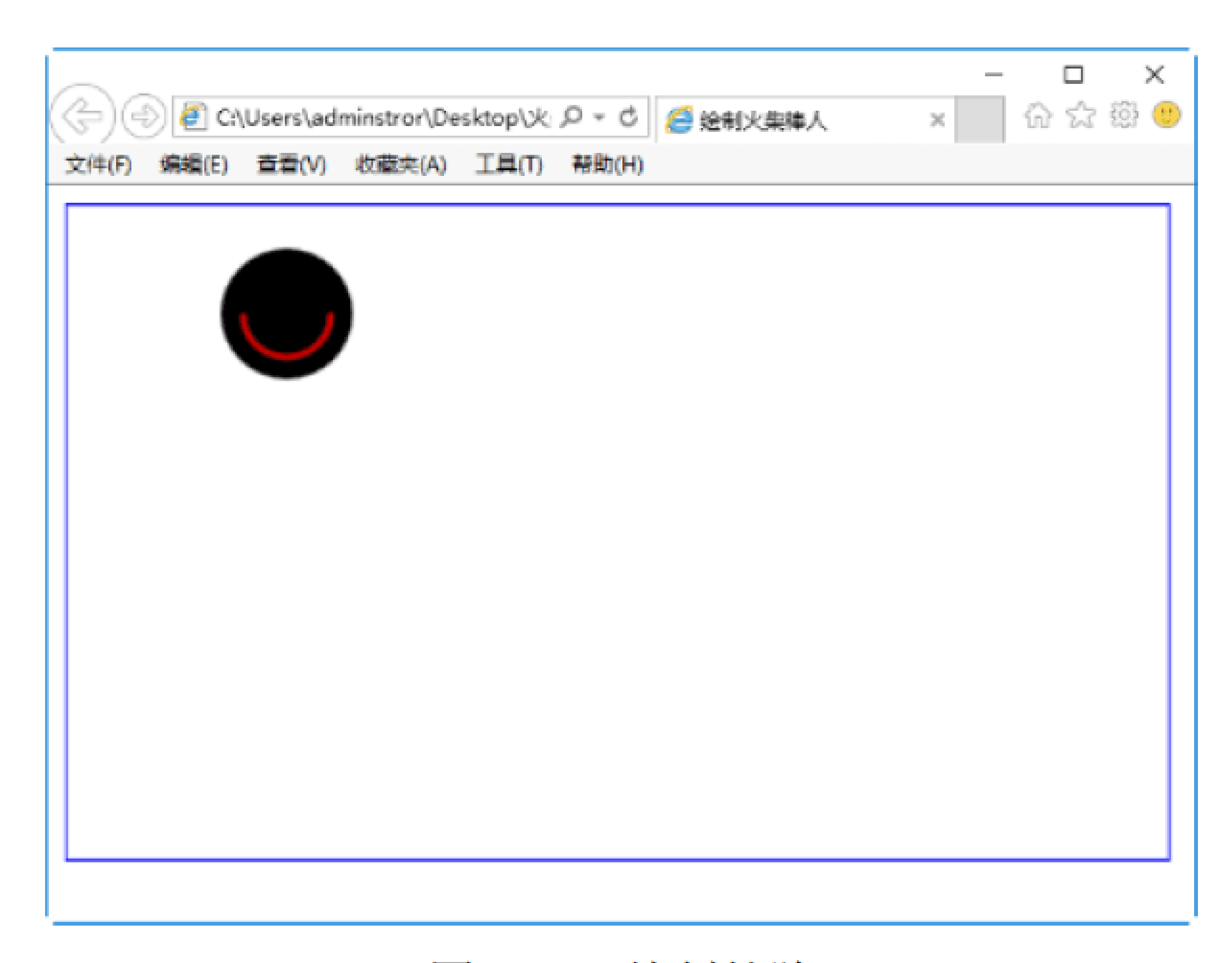


图 7-23　绘制笑脸

05 JS 绘制眼睛。

```
cxt.beginPath();
cxt.fillStyle="#c00";
cxt.arc(90,45,3,0,Math.PI*2,true);
cxt.fill();
cxt.moveTo(113,45);
cxt.arc(110,45,3,0,Math.PI*2,true);
cxt.fill();
cxt.stroke();
```

首先填充弧线，创建一个实体样式的眼睛，使用 arc 绘制左眼，然后使用 moveto 绘制右眼。在 IE 11.0 中浏览效果如图 7-24 所示，页面上显示了一双眼睛。

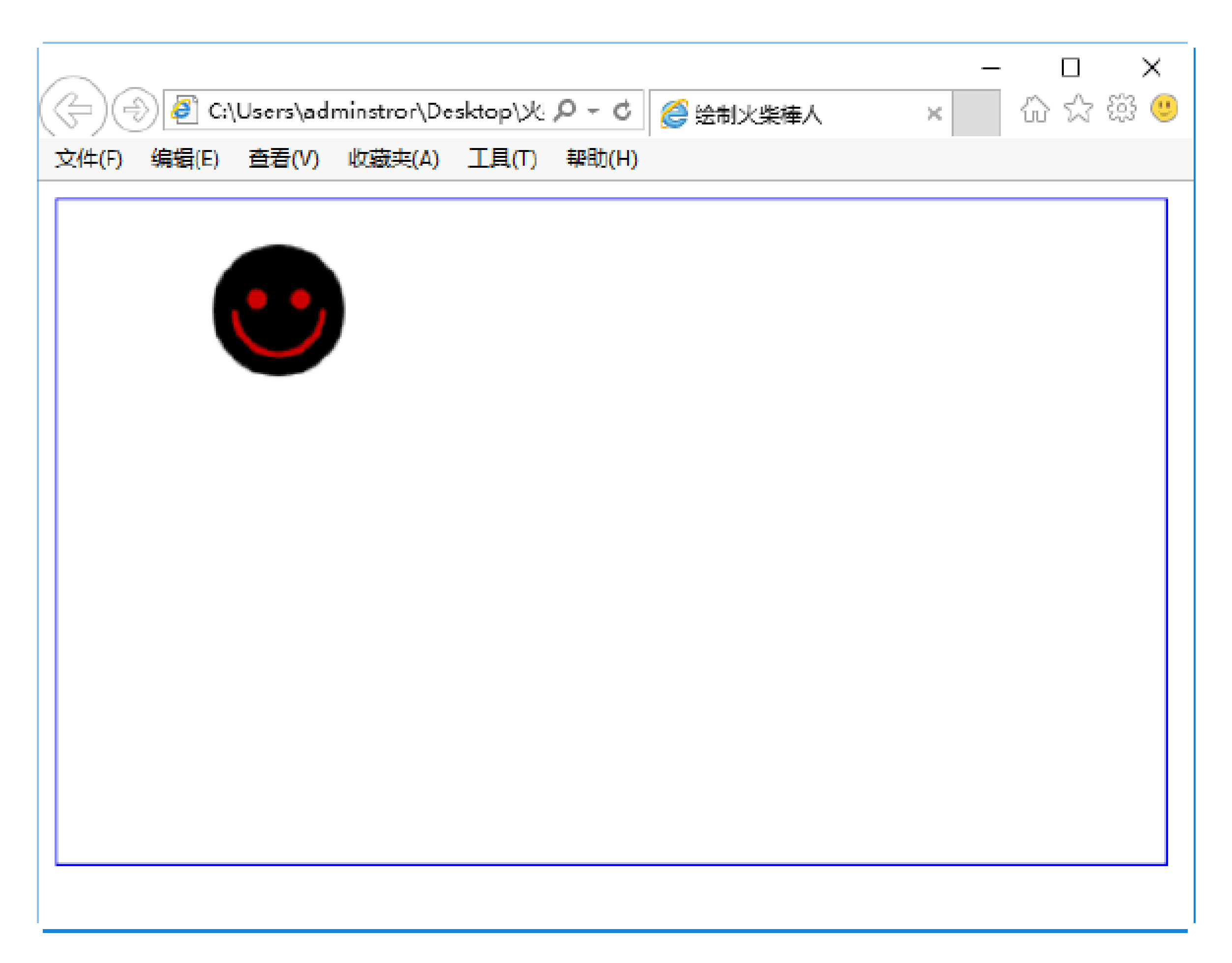


图 7-24 绘制眼睛

06 绘制身躯。

```
cxt.moveTo(100,80);
cxt.lineTo(100,150);
cxt.moveTo(100,100),
cxt.lineTo(60,120);
cxt.moveTo(100,100);
cxt.lineTo(140,120);
cxt.moveTo(100,150);
cxt.lineTo(80,190);
cxt.moveTo(100,150);
cxt.lineTo(140,190);
cxt.stroke();
```

上面代码以 moveTo 作为开始坐标，以 lineTo 作为终点绘制不同的直线，这些直线的坐标位置需要在不同地方汇集，两只手在坐标位置（100，100）交叉，两只脚在坐标位置（100,150）交叉。

在 IE 11.0 中浏览效果如图 7-25 所示，页面显示了一个火柴棒人。

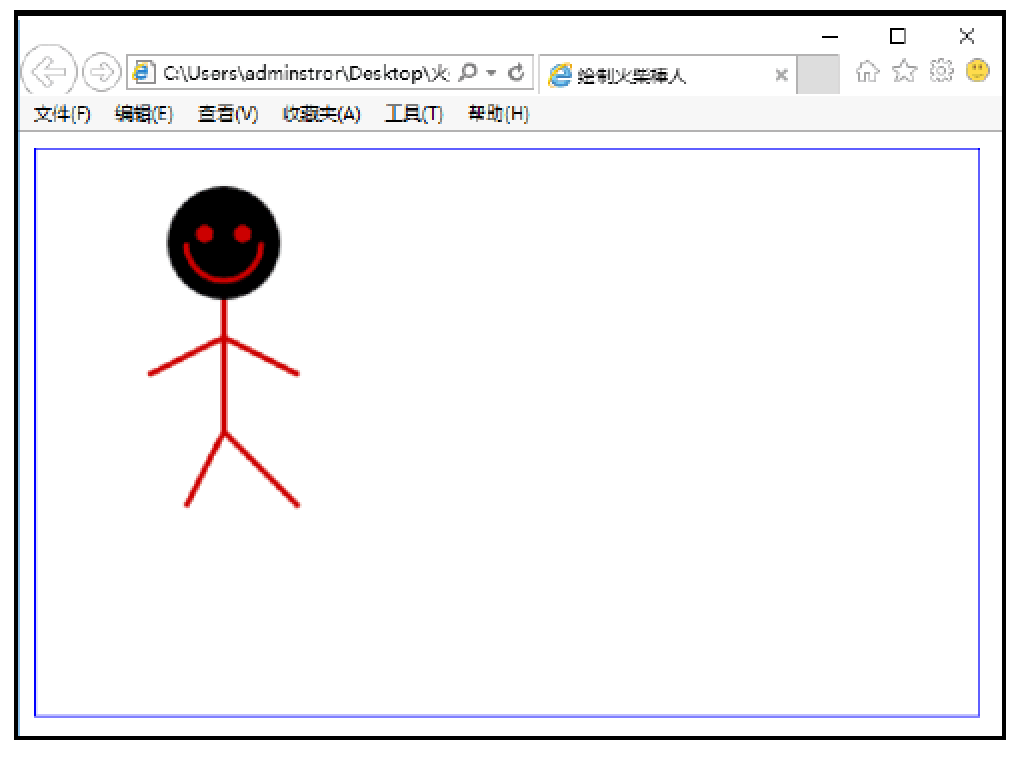


图 7-25 定义身躯

7.12 综合实例 3—绘制时钟

利用容器画布 canvas 这个新的特性，可以在网页上创建一个类似于钟表的特效。具体步骤如下：

01 分析需求。

在画布上绘制时钟，需要绘制几个必要的图形：表盘、时针、分针、秒针和中心圆，这样将这几个图形组合起来就构成一个时针界面，然后使用 JS 代码，根据时间定秒针、分针和时针。

02 创建 HTML 页面。

```
<!DOCTYPE html>
<html>
<head>
<title>canvas时钟</title>
</head>
<body>
<canvas id="canvas" width="200" height="200" style="border:1px solid
#000;">您的浏览器不支持Canvas。</canvas>
</body>
</html>
```

上面代码创建了一个画布，其宽度为 200 像素，高度为 200 像素，带有边框，颜色为黑色，样式为直线型。在 IE 11.0 中浏览效果如图 7-26 所示，可以看到显示了一个带有黑色边框的画布，画布中没有任何信息。

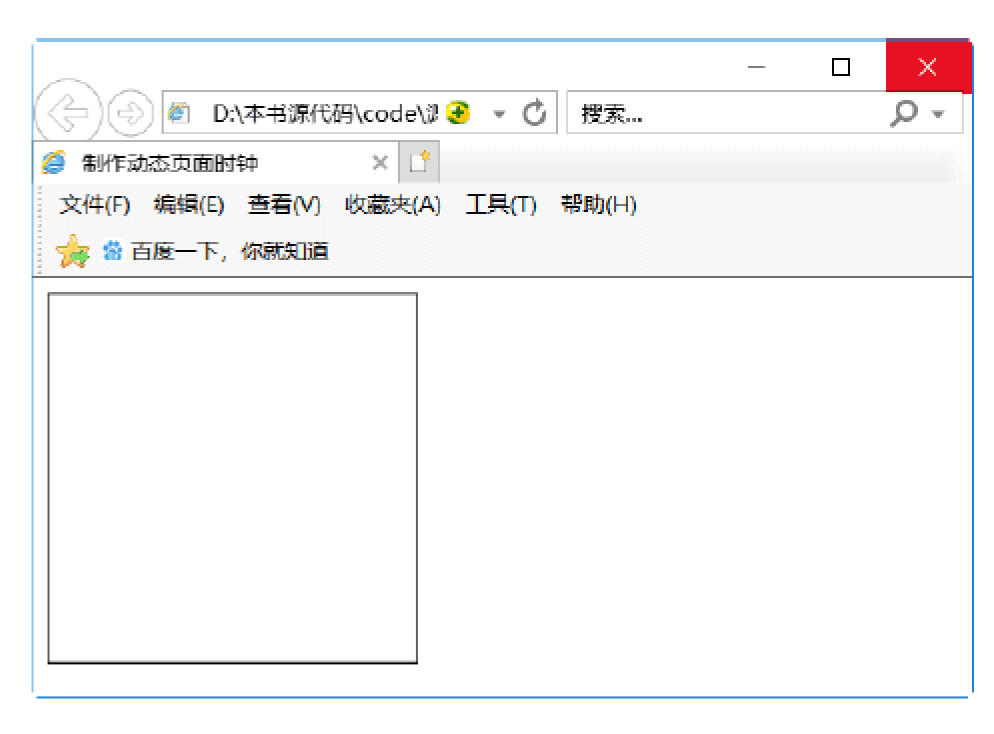


图 7-26 定义画布

03 添加 JavaScript，绘制不同的图形。

```
<script type="text/javascript" language="javascript" charset="utf-8">
 var canvas = document.getElementById('canvas');
 var ctx = canvas.getContext('2d');
 if(ctx){
```

```
var timerId;
var frameRate = 60;
function canvObject(){
 this.x = 0;
 this.y = 0;
 this.rotation = 0;
 this.borderWidth = 2;
 this.borderColor = '#000000';
 this.fill = false;
 this.fillColor = '#ff0000';
 this.update = function(){
  if(!this.ctx)throw new Error('您没有指定ctx对象。');
  var ctx = this.ctx
  ctx.save();
  ctx.lineWidth = this.borderWidth;
  ctx.strokeStyle = this.borderColor;
  ctx.fillStyle = this.fillColor;
  ctx.translate(this.x, this.y);
  if(this.rotation)ctx.rotate(this.rotation * Math.PI/180);
  if(this.draw)this.draw(ctx);
  if(this.fill)ctx.fill();
  ctx.stroke();
  ctx.restore();
 }
};
function Line(){};
Line.prototype = new canvObject();
Line.prototype.fill = false;
Line.prototype.start = [0,0];
Line.prototype.end = [5,5];
Line.prototype.draw = function(ctx){
 ctx.beginPath();
 ctx.moveTo.apply(ctx,this.start);
 ctx.lineTo.apply(ctx,this.end);
 ctx.closePath();
};

function Circle(){};
Circle.prototype = new canvObject();
Circle.prototype.draw = function(ctx){
 ctx.beginPath();
 ctx.arc(0, 0, this.radius, 0, 2 * Math.PI, true);
 ctx.closePath();
};

var circle = new Circle();
circle.ctx = ctx;
circle.x = 100;
circle.y = 100;
circle.radius = 90;
circle.fill = true;
circle.borderWidth = 6;
circle.fillColor = '#ffffff';
```

```
var hour = new Line();
hour.ctx = ctx;
hour.x = 100;
hour.y = 100;
hour.borderColor = "#000000";
hour.borderWidth = 10;
hour.rotation = 0;
hour.start = [0,20];
hour.end = [0,-50];

var minute = new Line();
minute.ctx = ctx;
minute.x = 100;
minute.y = 100;
minute.borderColor = "#333333";
minute.borderWidth = 7;
minute.rotation = 0;
minute.start = [0,20];
minute.end = [0,-70];

var seconds = new Line();
seconds.ctx = ctx;
seconds.x = 100;
seconds.y = 100;
seconds.borderColor = "#ff0000";
seconds.borderWidth = 4;
seconds.rotation = 0;
seconds.start = [0,20];
seconds.end = [0,-80];

var center = new Circle();
center.ctx = ctx;
center.x = 100;
center.y = 100;
center.radius = 5;
center.fill = true;
center.borderColor = ' green ';

for(var i=0,ls=[],cache;i<12;i++){
 cache = ls[i] = new Line();
 cache.ctx = ctx;
 cache.x = 100;
 cache.y = 100;
 cache.borderColor = " green ";
 cache.borderWidth = 2;
 cache.rotation = i * 30;
 cache.start = [0,-70];
 cache.end = [0,-80];
}

timerId = setInterval(function(){
 // 清除画布
```

```
        ctx.clearRect(0,0,200,200);
        // 填充背景色
        ctx.fillStyle = 'green';
        ctx.fillRect(0,0,200,200);
        // 表盘
        circle.update();
        // 刻度
        for(var i=0;cache=ls[i++];)cache.update();
        // 时针
        hour.rotation = (new Date()).getHours() * 30;
        hour.update();
        // 分针
        minute.rotation = (new Date()).getMinutes() * 6;
        minute.update();
        // 秒针
        seconds.rotation = (new Date()).getSeconds() * 6;
        seconds.update();
        // 中心圆
        center.update();
       },(1000/frameRate)|0);
     }else{
      alert('您的浏览器不支持Canvas无法预览时钟！');
     }
   </script>
```

上面代码由于篇幅比较长，只显示了部分代码。上面代码首先绘制不同类型的图形，如时针、秒针、分针等，然后将其组合在一起，并根据时间定义时针等指向。在 IE 11.0 中浏览效果如图 7-27 所示，可以看到页面中出现了一个时钟，其秒针在不停地移动。

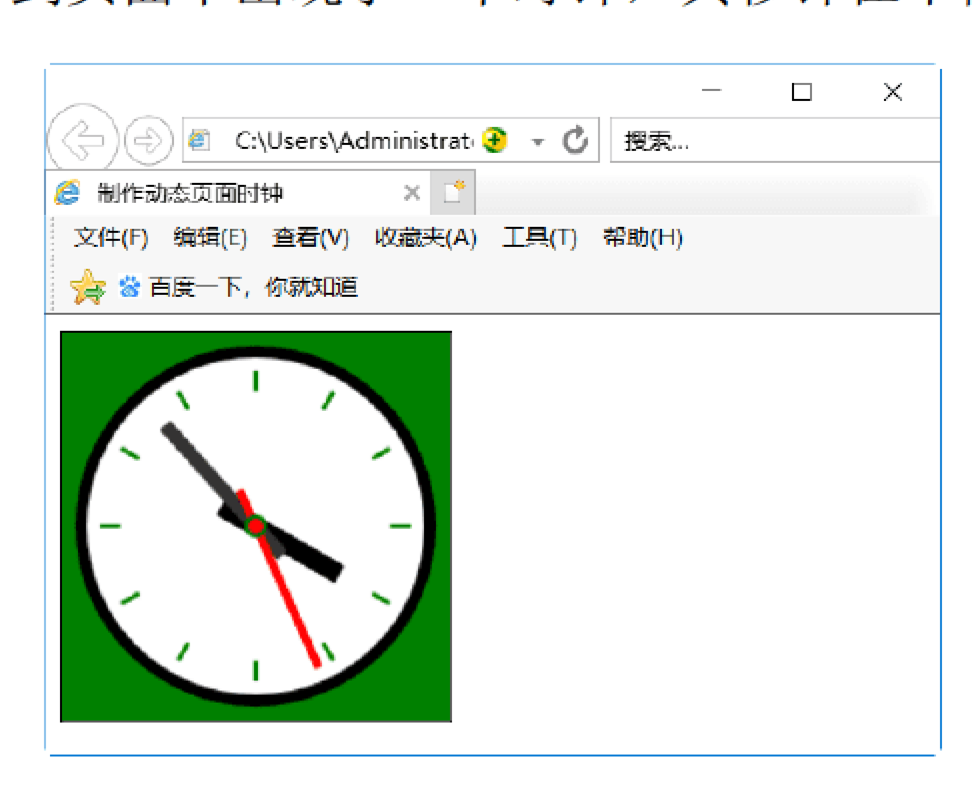


图 7-27　最终特效

7.13　专家解惑

1. 定义 canvas 宽度和高度时，是否可以在 CSS 属性中定义呢？

在添加一个 canvas 标记的时候，会在 canvas 的属性里填写要初始化的 canvas 的高度和宽

度：

```
<canvas width="500" height="400">Not Supported!</canvas>
```

如果把高度和宽度写在 css 里面，结果就会发现在绘图时坐标获取出现差异，canvas.width 和 canvas.height 分别是 300 和 150，和预期的不一样。这是因为 canvas 要求这两个属性必须和 canvas 标记一起出现。

2. 画布中 stroke 和 fill 二者的区别是什么？

HTML5 中将图形分为两大类：第一类称作 stroke，就是轮廓、勾勒或线条，总之图形是由线条组成的；第二类称作 fill，即填充区域。上下文对象中有两个绘制矩形的函数，可以让我们很好地理解这两大类图形的区别，即 strokeRect 和 fillRect。

第8章　地理定位、离线Web应用和Web存储

在 HTML5 中，由于地理定位、离线 Web 应用和 Web 存储技术的出现，用户可以查找网站访问者的当前位置。在线时可以快速地存储网站的相关信息，当用户再次访问网站时，将大大提升访问的速度，即使网站脱机，仍然可以访问站点。

8.1　获取地理位置

在 HTML5 网页代码中，通过一些有用的 API，可以查找访问者当前的位置。下面将详细讲述地理位置获取的方法。

8.1.1　地理地位的原理

由于访问者浏览网站的方式不用，可以通过下列方式确定其位置。

（1）如果网站浏览者使用电脑上网，则通过获取浏览者的 IP 地址，从而确定其具体位置。

（2）如果网站浏览者通过手机上网，则通过获取浏览者的手机信号接收塔，从而确定其具体位置。

（3）如果网站浏览者的设备上具有 GPS 硬件，则通过获取 GPS 发出的载波信号，从而确定其具体位置。

（4）如果网站浏览者通过无线上网，则通过无线网络连接获取其具体位置。



API 是应用程序的编程接口，是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件以访问一组例程的能力。

8.1.2 地理定位的函数

通过地理定位可以确定用户的当前位置，并能获取用户地理位置的变化情况。其中，比较常用的就是 API 中的 getCurrentpositong 函数。

getCurrentpositong 函数的语法格式如下：

```
void getCurrentPosition(successCallback, errorCallback, options);
```

其中 successCallback 参数是指在位置成功获取时用户想要调用的函数名称；errorCallback 参数是指在位置获取失败时用户想要调用的函数名称；options 参数指出地理定位时的属性设置。



访问用户位置是耗时的操作，同时出于隐私问题，还要取得用户的同意。

如果地理定位成功，则新的 Position 对象将调用 displayOnMap 函数，显示设备的当前位置。

那么 Positon 对象的含义是什么呢？作为地理定位的 API，Positon 对象包含位置确定时的时间戳（timestamp）和包含位置的坐标（coords），具体语法格式如下：

```
Interface position
 {
readonly attribute Coordinates cords;
readonly attribute DOMTimeStamp timestamp;
};
```

8.1.3 指定纬度和经度坐标

地理定位成功后，将调用 displayOnMap 函数。此函数如下：

```
function displayOnMap(position)
{
var latitude=positon.coords.latitude;
var longitude=postion.coords.longitude;
}
```

其中第 1 行函数从 Position 对象获取 coordinates 对象，主要由 API 传递给程序调用；第 3 行和第 4 行中定义了两个变量，latitude 和 longitude 属性存储在定义的两个变量中。

为了在地图上显示用户的具体位置，可以利用地图网站的 API。下面以使用百度地图为例进行讲解，需要使用 Baidu Maps Javascript API。在使用此 API 前，需要在 HTML5 页面中添加一个引用，具体代码如下：

```
<--baidu maps API>
```

```
    <script type=“text/javascript” scr=
“http://api.map.baidu.com/api?key=*&v=1.0&services=true” >
    </script>
```

其中*号代码注册到 key。注册 key 的方法为：在“http：//openapi.baidu.com/map/index.html”网页中注册百度地图 API，然后输入需要内置百度地图页面的 URL 地址生成 API 密钥，最后将 key 文件复制保存。

虽然已经包含了 Baidu Maps Javascript，但是页面中还不能显示内置的百度地图，还需要添加 html 语言。使地图从程序转化为对象需要加入以下源代码：

```
    <script
type="text/javascript"scr="http://api.map.baidu.com/api?key=*&v=1.0&service
s=true">
    </script>
    <div style="width:600px;height:220px;border:1px solid
gary;margin-top:15px;" id="container">
    </div>
    <script type="text/javascript">
    var map=new BMap.Map("container");
    map.centerAndZoom(new BMap.Point(***,***),17);
    map.addControl(new BMap.NavigationControl());
    map.addControl(new BMap.ScaleControl());
    map.addControl(new BMap.OverviewMapControl());
    var local=new BMap.LocalSearch(map,
    {
    enderOptions:{map: map}
    }
    );
    local.search("输入搜索地址");
    </script>
```

上述代码分析如下：

（1）其中前 2 行主要是把 baidu map API 程序植入源码中。

（2）第 3 行在页面中设置一个标记，包括宽度和长度，用户可以自己调整；border=1px 是定义外框的宽度为 1 个像素，solid 为实线，gray 为边框显示颜色，margin-top 为该标记距离与上部的距离。

（3）第 7 行为地图中自己位置的坐标。

（4）第 8~10 行为植入地图缩放控制工具。

（5）第 11~16 行为地图中自己的位置，只需在 local search 后填入自己的位置名称即可。

8.1.4 目前浏览器对地理定位的支持情况

不同的浏览器版本对地理定位技术的支持情况也是不同的。表 8-1 是常见浏览器对地理定位的支持情况。

表 8-1 浏览器支持情况

浏览器名称	支持 Web 存储技术的版本
Internet Explorer	Internet Explorer 9 及更高版本
Firefox	Firefox 3.5 及更高版本
Opera	Opera 10.6 及更高版本
Safari	Safari 5 及更高版本
Chrome	Chrome 5 及更高版本
Android	Android 2.1 及更高版本

8.2 HTML5 离线 Web 应用

为了能在离线的情况下访问网站，可以采用 HTML5 的离线 Web 功能。下面来学习 Web 应用程序如何缓存。

8.2.1 新增的本地缓存

在 HTML5 中新增了本地缓存（也就是 HTML 离线 Web 应用），主要是通过应用程序缓存整个离线网站的 HTML、CSS、Javascript、网站图像和资源。当服务器没有和 Internet 建立连接的时候，也可以利用本地缓存中的资源文件来正常运行 Web 应用程序。

另外，如果网站发生了变化，则应用程序缓存将重新加载变化的数据文件。

8.2.2 本地缓存的管理者——manifest 文件

客户端的浏览器是如何知道应该缓存哪些文件的呢？这就需要依靠 manifest 文件来管理。manifest 文件是一个简单文本文件，在该文件中以清单的形式列举了需要被缓存或不需要被缓存的资源文件的文件名称，以及这些资源文件的访问路径。

Manifest 文件把指定的资源文件类型分为三类，分别是“CACHE”“NETWORK”和“FALLBACK”。这三类的含义分别如下：

（1）CACHE 类别：该类别指定需要被缓存在本地的资源文件。这里需要特别注意的是，在为某个页面指定需要本地缓存的资源文件时，不需要把这个页面本身指定在 CACHE 类型中，

因为如果一个页面具有 manifest 文件，则浏览器会自动对这个页面进行本地缓存。

（2）NETWORK 类别：该类别为不进行本地缓存的资源文件，这些资源文件只有当客户端与服务器端建立连接的时候才能访问。

（3）FALLBACK 类别：该类别中指定两个资源文件，其中一个资源文件为能够在线访问时使用的资源文件，另一个资源文件为不能在线访问时使用的备用资源文件。

以下是一个简单的 manifest 文件的内容。

```
CACHE MANIFEST
#文件的开头必须是CACHE MANIFEST
CACHE:
123.html
myphoto.jpg
12.php
NETWORK:
http://www.baidu.com/xxx
feifei.php
FALLBACK:
online.js locale.js
```

上述代码含义分析如下：

（1）指定资源文件，文件路径可以是相对路径，也可以是绝对路径。指定时每个资源文件为独立的一行。

（2）第一行必须是 CACHE MANIFEST，此行的作用告诉浏览器需要对本地缓存中的资源文件进行具体设置。

（3）每一个类型都是必须出现的，而且同一个类别可以重复出现。如果文件开头没有指定类别而直接书写资源文件，则浏览器会把这些资源文件视为 CACHE 类别。

（4）在 manifest 文件中，注释行以“#”开始，主要用于进行一些必要的说明或解释。

为单个网页添加 manifest 文件时，需要在 Web 应用程序页面上的 html 元素的 manifest 属性中指定 manifest 文件的 URL 地址。具体的代码如下：

```
<html manifest="123.manifest">
</html>
```

添加上述代码后，浏览器就能够正常地阅读该文本文件。



用户可以为每一个页面单独指定一个 mainifest 文件，也可以对整个 Web 应用程序指定一个总的 manifest 文件。

上述操作完成后，即可实现资源文件缓存到本地。当要对本地缓存区的内容进行修改时，只需要修改 manifest 文件。文件被修改后，浏览器可以自动检查 manifest 文件，并自动更新本地缓存区中的内容。

8.2.3 浏览器网页缓存与本地缓存的区别

浏览器网页缓存与本地缓存的主要区别如下：

（1）因为浏览器网页缓存主要是为了加快网页加载的速度，所以会对每一个打开的网页都进行缓存操作，而本地缓存是为整个 Web 应用程序服务的，只缓存那些指定缓存的网页。

（2）在网络连接的情况下，浏览器网页缓存一个页面的所有文件，但是一旦离线，用户单击链接时，将会得到一个错误消息。而本地缓存在离线时，仍然可以正常访问。

（3）对于网页浏览者而言，浏览器网页缓存了哪些内容和资源，这些内容是否安全可靠等等都不知道，而本地缓存的页面是编程人员指定的内容，在安全方面相对可靠许多。

8.2.4 目前浏览器对 Web 离线应用的支持情况

不同的浏览器版本对 Web 离线应用技术的支持情况也是不同的。表 8-2 是常见浏览器对 Web 离线应用的支持情况。

表 8-2　浏览器支持情况

浏览器名称	支持 Web 存储技术的版本情况
Internet Explorer	Internet Explorer 9 及更低版本目前尚不支持
Firefox	Firefox 3.5 及更高版本
Opera	Opera 10.6 及更高版本
Safari	Safari 4 及更高版本
Chrome	Chrome 5 及更高版本
Android	Android 2.0 及更高版本

8.3 Web 存储

在 HTML5 标准之前，Web 存储信息需要 Cookies 来完成，但是 Cookie 不适合大量数据的存储，因为它们由每个对服务器的请求来传递，这使得 Cookies 速度很慢而且效率也不高。为此，在 THML5 中，Web 存储 API 为用户如何在计算机或设备上存储用户信息作了数据标准的定义。

8.3.1 本地存储和 Cookies 的区别

本地存储和 Cookies 扮演着类似的角色，但是它们有根本的区别。

（1）本地存储是仅存储在用户的硬盘上并等待用户读取，而 Cookies 是在服务器上读取。

（2）本地存储仅供客户端使用，如果需要服务器端根据存储数值作出反映，就应该使用 Cookies。

（3）读取本地存储不会影响到网络带宽，但是使用 Cookies 将会发送到服务器，这样会影响到网络带宽，无形中增加了成本。

（4）从存储容量上看，本地存储可存储多达 5MB 的数据，而 Cookies 最多只能存储 4KB 的数据信息。

8.3.2 在客户端存储数据

在 HTML5 标准中，提供了以下两种在客户端存储数据的新函数。

（1）sessionStorage：针对一个 session 的数据存储，也被称为会话存储。让用户跟踪特定窗口中的数据，即使同时打开两个窗口是同一站点，每个窗口也有自己独立的存储对象，但是用户会话的持续时间只是限定在用户打开浏览器窗口的时间，一旦关闭浏览器窗口，用户会话将结束。

（2）localStorage：没有时间限制的数据存储，也被称为本地存储，和会话存储不用，本地存储将在用户计算机上永久保持数据信息。关闭浏览器窗口后，如果再次打开该站点，就可以检索所有存储在本地上的数据。

在 HTML5 中，数据不是由每个服务器请求传递的，而是只有在请求时使用数据，这样的话，存储大量数据时不会影响网站性能。对于不同的网站，数据存储于不同的区域，并且一个网站只能访问其自身的数据。



HTML5 使用 JavaScript 来存储和访问数据，建议用户可以多了解一下 JavaScript 的基本知识。

8.3.3 sessionStorage 函数

sessionStorage 函数针对一个 session 进行数据存储。如果用户关闭浏览器窗口，则数据会被自动删除。

创建一个 sessionStorage 函数的基本语法格式如下：

```
<script type="text/javascript">
sessionStorage.abc="  ";
</script>
```

【例 8.1】（实例文件：ch8\8.1.html）

```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
sessionStorage.name="我们的公司是:英达科技文化公司";
document.write(sessionStorage.name);
</script>
</body>
</html>
```

在 Firefox 62.0 中浏览效果如图 8-1 所示，可以看到 sessionStorage 函数创建的对象内容已经显示在网页中。

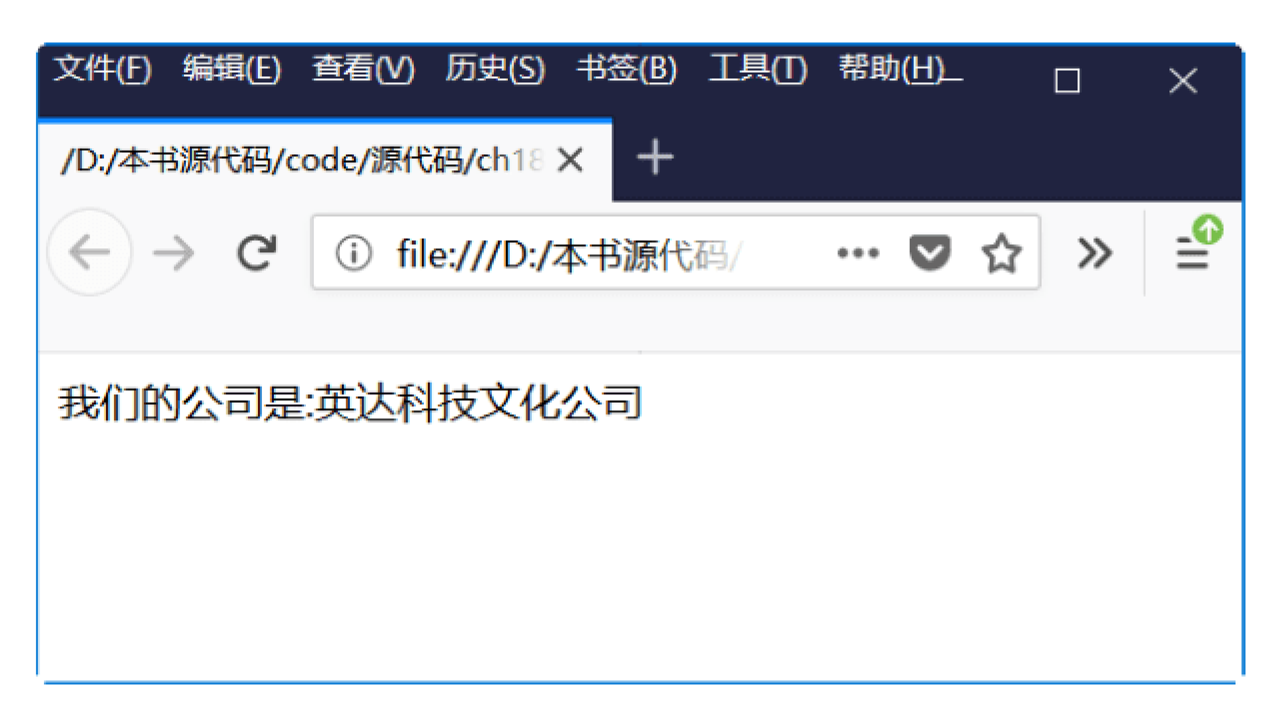


图 8-1　sessionStorage 函数创建对象的效果

下面继续使用 sessionStorage 函数来做一个实例，主要制作记录用户访问网站次数的计数器。

【例 8.2】（实例文件：ch8\8.2.html）

```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
if(sessionStorage.count)
{
sessionStorage.count=Number(sessionStorage.count)+1;
}
else
{
sessionStorage.count=1;
}
document.write("您访问该网站的次数为："+sessionStorage.count);
</script>
```

```
</body>
</html>
```

在 Firefox 62.0 中浏览效果如图 8-2 所示。如果用户刷新一次页面，则计数器的数值会进行加 1。

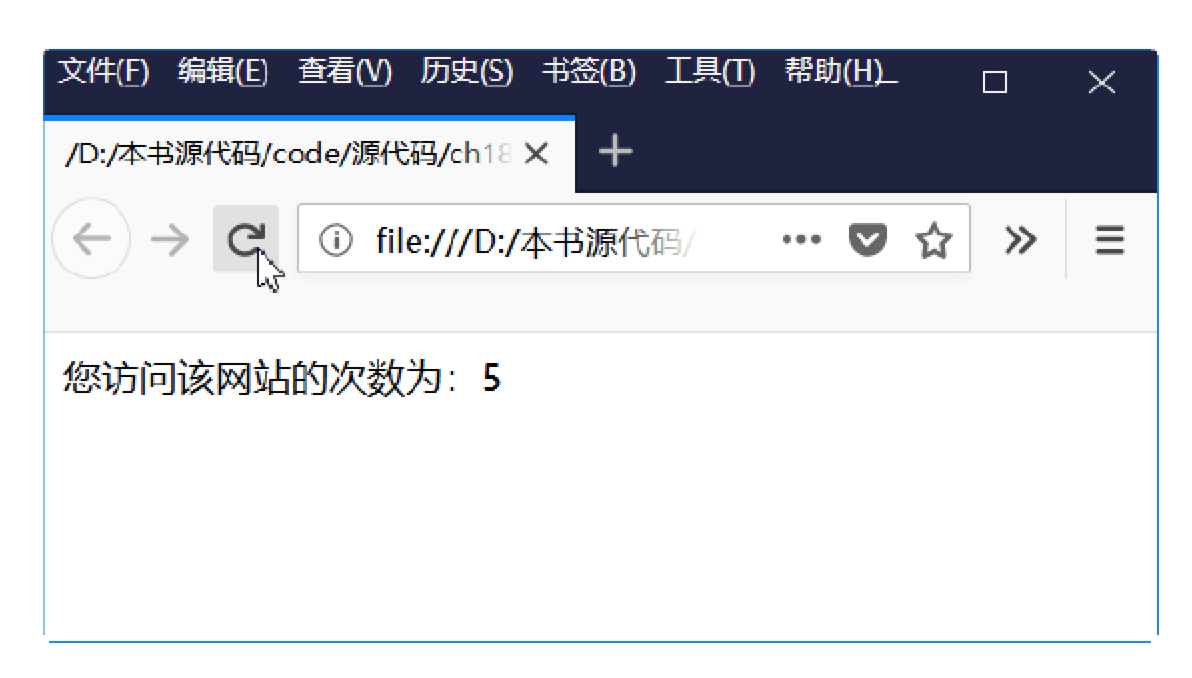


图 8-2　sessionStorage 方法创建计数器效果



如果用户关闭浏览器窗口，再次打开该网页，则计数器将重置为 1。

8.3.4　localStorage 函数

与 seessionStorage 函数不同，localStorage 函数存储的数据没有时间限制。也就是说网页浏览者关闭网页很长一段时间后，再次打开此网页时，数据依然可用。

创建一个 localStorage 函数的基本语法格式如下：

```
<script type="text/javascript">
localStorage.abc="  ";
</script>
```

【例 8.3】（实例文件：ch8\8.3.html）

```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
localStorage.name="学习HTML5最新的技术：Web存储";
document.write(localStorage.name);
</script>
</body>
</html>
```

在 Firefox 62.0 中浏览效果如图 8-3 所示，可以看到 localStorage 函数创建的对象内容显示在网页中。

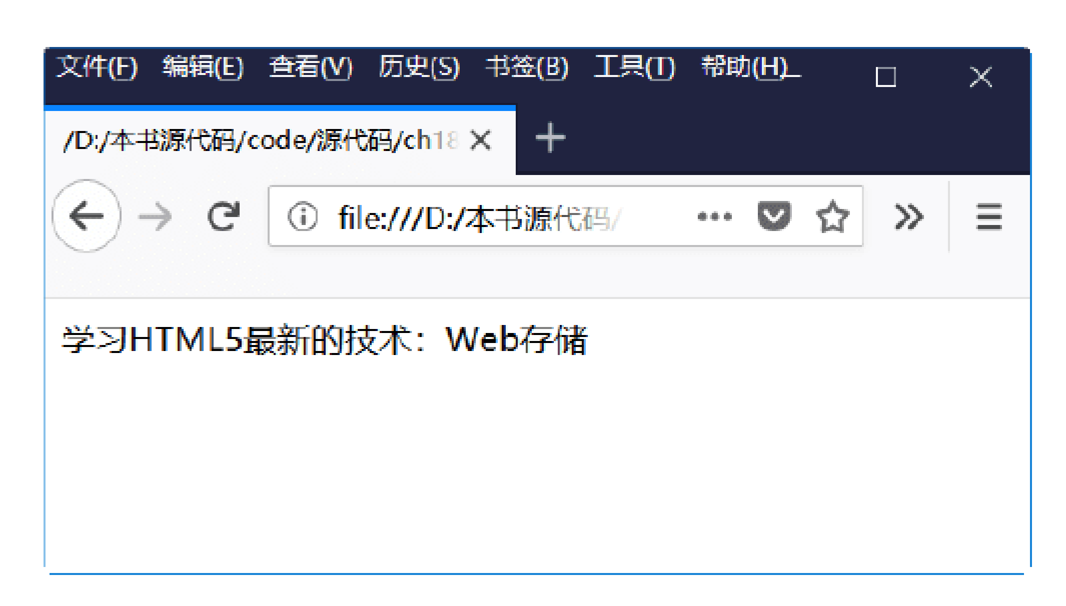


图 8-3　localStorage 函数创建对象的效果

下面仍然使用 localStorage 函数来制作记录用户访问网站次数的计数器。用户可以清楚地看到 localStorage 函数和 sessionStorage 函数的区别。

【例 8.4】（实例文件：ch8\8.4.html）

```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
if (localStorage.count)
{
localStorage.count=Number(localStorage.count)+1;
}
else
{
localStorage.count=1;
}
document.write("您访问该网站的次数为: "+localStorage.count");
</script>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 8-4 所示。如果用户刷新一次页面，则计数器的数值将进行加 1；如果用户关闭浏览器窗口，再次打开该网页，则计数器会继续上一次计数，而不会重置为 1。

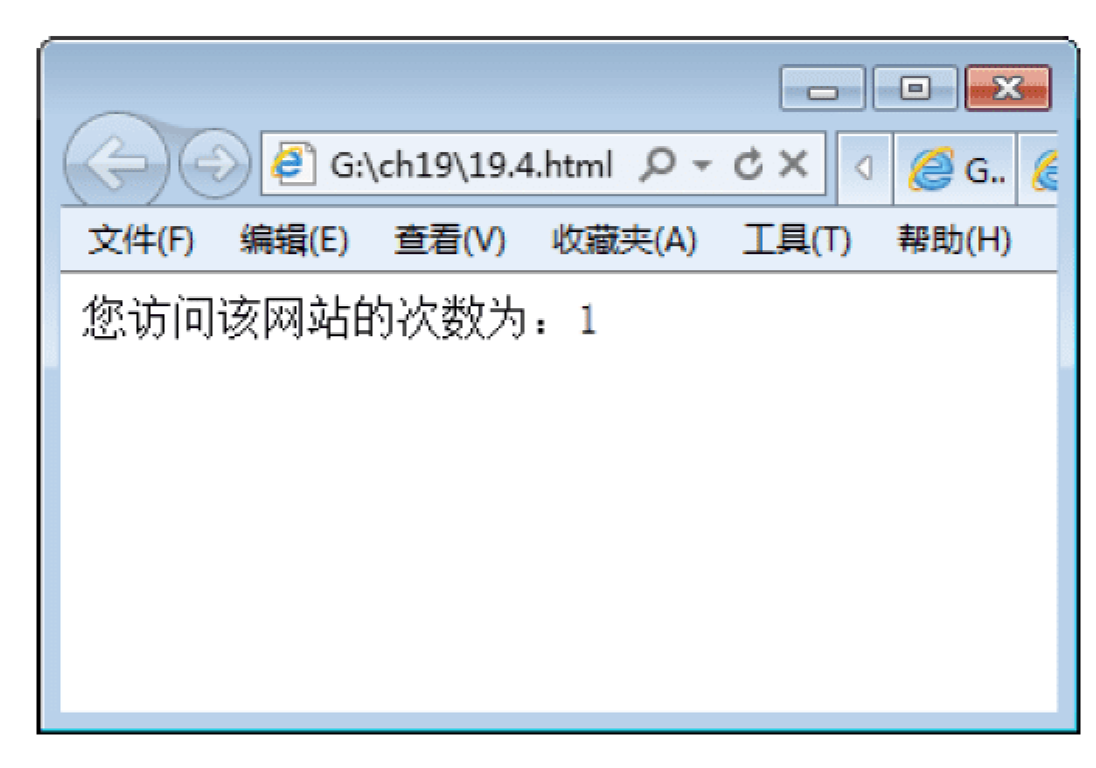


图 8-4　localStorage 函数创建计数器效果

8.3.5 目前浏览器对 Web 存储的支持情况

不同的浏览器版本对 Web 存储技术的支持情况也是不同的。表 8-3 是常见浏览器对 Web 存储的支持情况。

表 8-3 浏览器支持情况

浏览器名称	支持 Web 存储技术的版本
Internet Explorer	Internet Explorer 8 及更高版本
Firefox	Firefox 3.6 及更高版本
Opera	Opera 10.0 及更高版本
Safari	Safari 4 及更高版本
Chrome	Chrome 5 及更高版本
Android	Android 2.1 及更高版本

8.4 专家解惑

1. 不同的浏览器可以读取同一个 Web 中存储的数据吗？

在 Web 存储时，不同的浏览器将存储在不同的 Web 存储库中。例如，如果用户使用的是 IE 浏览器，那么 Web 存储工作时，所有数据将存储在 IE 的 Web 存储库中；如果用户再次使用火狐浏览器访问该站点，就不能读取 IE 浏览器存储的数据，可见每个浏览器的存储是分开并独立工作的。

2. 离线存储站点时是否需要浏览者同意？

和地理定位类似，在网站使用 manifest 文件时，浏览器会提供一个权限提示，提示用户是否将离线设为可用，但是不是每一个浏览器都支持这样的操作。

第9章　CSS3快速入门

一个美观大方简约的页面以及高访问量的网站，是网页设计者的追求。然而，仅通过HTML5 实现是非常困难的，HTML 语言仅仅定义了网页结构，对于文本样式而没有过多涉及。这就需要一种技术对页面布局、字体、颜色、背景和其他图文效果的实现提供更加精确地控制，这种技术就是 CSS。

9.1　CSS3 介绍

CSS3 最大的优势是在后期维护中，如果一些外观样式需要修改，就只修改相应的代码即可。

9.1.1　CSS3 功能

随着 Internet 不断地发展，对页面效果诉求越来越强烈，只依赖 HTML 这种结构化标记来实现样式已经不能满足网页设计者的需要，其表现在以下几个方面。

（1）维护困难。为了修改某个特殊标记格式，需要花费很多时间，尤其对整个网站而言，后期修改和维护成本较高。

（2）标记不足。HTML 本身标记并不是很多，而且很多标记都是为网页内容服务的，关于内容样式的标记（如文字间距、段落缩进）很难在 HTML 中找到。

（3）网页过于臃肿。由于没有统一对各种风格样式进行控制，HTML 页面往往体积过大，占用掉很多宝贵的宽度。

（4）定位困难。在整体布局页面时，HTML 对于各个模块的位置调整显得捉襟见肘，过多的<table>标记将会导致页面的复杂和后期维护的困难。

在这种情况下，就需要寻找一种可以将结构化标记与丰富页面表现相结合的技术，而 CSS 样式技术恰恰迎合了这种需要。

CSS（Cascading Style Sheet）称为层叠样式表，也可以称为 CSS 样式表或样式表，其文件

扩展名为.css。CSS 是用于增强或控制网页样式，并允许将样式信息与网页内容分离的一种标记性语言。

引用样式表的目的是将“网页结构代码”和“网页样式风格代码”分离开，从而使网页设计者可以对网页布局进行更多的控制。利用样式表可以将整个站点上所有网页都指向某个 CSS 文件，设计者只需要修改 CSS 文件中的某一行，整个网页上对应的样式就会随之发生改变。

9.1.2 CSS3 发展历史

万维网联盟（W3C）在 1996 年制定并发布了一个网页排版样式标准（层叠样式表）用来对 HTML 有限的表现功能进行补充。

随着 CSS 的广泛应用，CSS 技术越来越成熟。CSS 现在有三个不同层次的标准：CSS1、CSS2 和 CSS3。

CSS1（CSS Level 1）是 CSS 的第一层次标准，它正式发布于 1996 年 12 月 17 日，后来于 1999 年 1 月 11 日进行了修改。该标准提供简单的样式表机制，使网页的设计者可以通过附属样式对 HTML 文档的表现进行描述。

CSS2（CSS Level 2）于 1998 年 5 月 12 日被正式作为标准发布。CSS2 标准是基于 CSS1 设计的，其包含了 CSS1 所有的功能，并扩充和改进了很多更加强大的属性。CSS2 支持多媒体样式表，使得设计者可以根据不同的输出设备给文档制定不同的表现形式。

在 2001 年 5 月 23 日，W3C 完成了 CSS3 的工作草案。该草案制订了 CSS3 的发展路线图，详细列出了所有模块，并计划在未来进行逐步规范。

CSS 1 主要定义了网页的基本属性，如字体、颜色、空白边等。CSS 2 在此基础上添加了一些高级功能（如浮动和定位），以及一些高级的选择器（如子选择器、相邻选择器和通用选择器等）。CSS3 开始遵循模块化开发，标准被分为若干个相互独立的模块，这将有助于理清模块化规范之间的关系，减小完整文件的体积。

9.1.3 浏览器与 CSS3

CSS3 制定完成之后具有了很多新功能（新样式），但这些新样式在浏览器中不能获得完全支持。主要在于各个浏览器对 CSS3 很多细节处理上存在差异，例如某个标记属性一种浏览器支持而另一种浏览器不支持，或者两者浏览器都支持但其显示效果不一样。

主流浏览器为了自己产品利益和推广，定义了很多私有属性用于加强页面显示样式和效果，导致现在每个浏览器都存在大量的私有属性。虽然使用私有属性可以快速构建效果，但是对网页设计者是一个大麻烦。设计一个页面，就需要考虑在不同浏览器上的显示效果，一个不注意就会导致同一个页面在不同浏览器上显示效果不一致。甚至有的浏览器不同版本之间，也

具有不同的属性。

如果所有浏览器都支持 CSS3 样式，那么网页设计者只需使用一种统一标记，即可在不同浏览器上实现一致的显示效果。

当 CSS3 被所有浏览器接受和支持以后，整个网页设计将会变得非常容易。CSS3 标准使得布局更加合理，样式更加美观，整个 Web 页面显示将会焕然一新。虽然现在 CSS3 还没有完全普及，各个浏览器对 CSS3 的支持还处于发展阶段，但 CSS3 具有很高的发展潜力，在样式修饰方面是其他技术无法替代的。此时学习 CSS3 技术，才能保证技术不落伍。

9.2 编辑和浏览 CSS

CSS 文件是文本格式文件，因此在编辑 CSS 时就有了多种选择，可以使用一些简单的文本编辑工具（如记事本、Word），也可以选择专业的 CSS 编辑工具（如 Dreamweaver）。记事本编辑工具适合于初学者，不适合大项目编辑，但专业工具软件通常占用空间较大，打开不太方便。

9.2.1 CSS 基础语法

在前面介绍过，CSS 样式表由若干条样式规则组成，这些样式规则可以应用到不同的元素或文档中来定义它们的显示效果。每一条样式规则由三部分构成：选择符（selector）、属性（property）和属性值（value）。其基本格式如下：

```
selector{property: value}
```

（1）selector 可以采用多种形式，可以为文档中的 HTML 标记（如<body>、<table>、<p>等），也可以是 XML 文档中的标记。

（2）property 是选择符指定的标记所包含的属性。

（3）value 指定了属性的值。如果定义选择符的多个属性，则属性和属性值为一组，组与组之间用“;”隔开。基本格式如下：

```
selector{property1: value1; property2: value2;…… }
```

下面就给出一条样式规则，代码如下：

```
p{color:red}
```

该样式规则的构成为：p 为段落提供样式，color 指定文字颜色属性，red 为属性值。此样式规则表示标记<p>指定的段落文字为红色。

如果要为段落设置多种样式，则可以使用下列语句。

```
p{font-family:"隶书"; color:red; font-size:40px; font-weight:bold}
```

9.2.2 使用记事本手工编写 CSS 文件

由于 CSS 是文本格式，因此使用传统的文本编辑器就可以编辑 CSS。当然这些编辑软件不支持语法提示，不支持验证，会严重影响开发效率，但使用记事本手工编写 CSS 文件可以使初学者更快地掌握 CSS3 技术。

使用记事本编写 CSS 和使用记事本编写 HTML 的方法一样。首先需要打开一个记事本，然后在里面输入相应 CSS 代码即可。

【例 9.1】使用记事本手工编写 CSS 文件。

01 打开记事本，编写 HTML 文档。

打开一个记事本，输入 HTML 代码，如图 9-1 所示。

02 添加 CSS 代码，修饰 HTML 元素。

在 head 部分添加 CSS 样式代码，效果如图 9-2 所示。

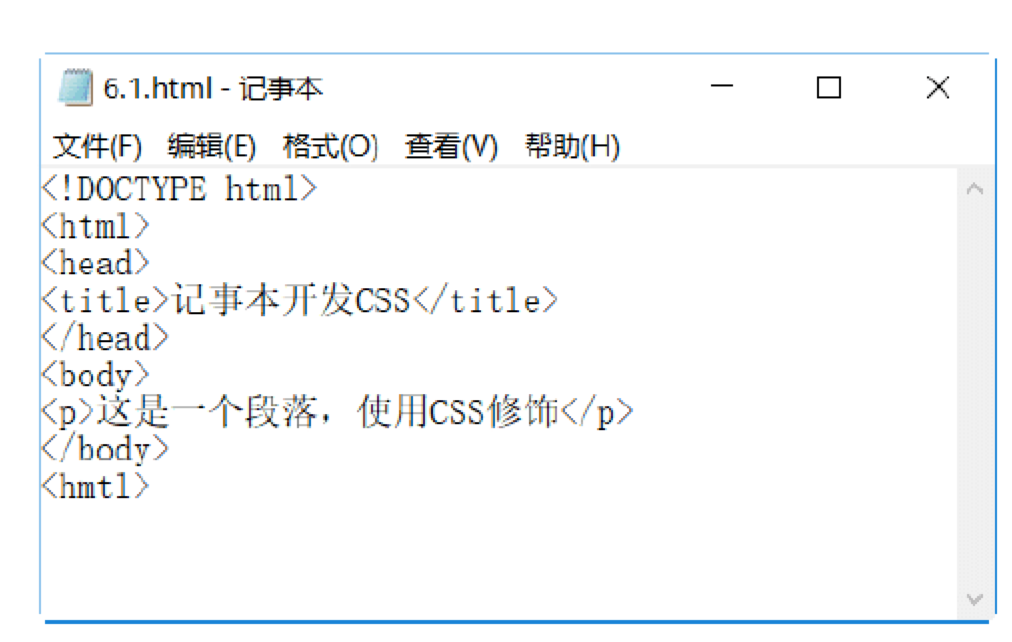


图 9-1 记事本开发 HTML

图 9-2 添加样式后效果

从窗口中可以看出，在 head 部分添加了一个<style>标记（即 CSS 样式标记）。在<style>标记中间对 p 样式进行了设置，设置段落居中显示并且颜色为红色。

03 运行网页文件。

编辑完成后，使用 IE 11.0 打开（如图 9-3 所示），可以看到段落在页面中间以红色字体显示。

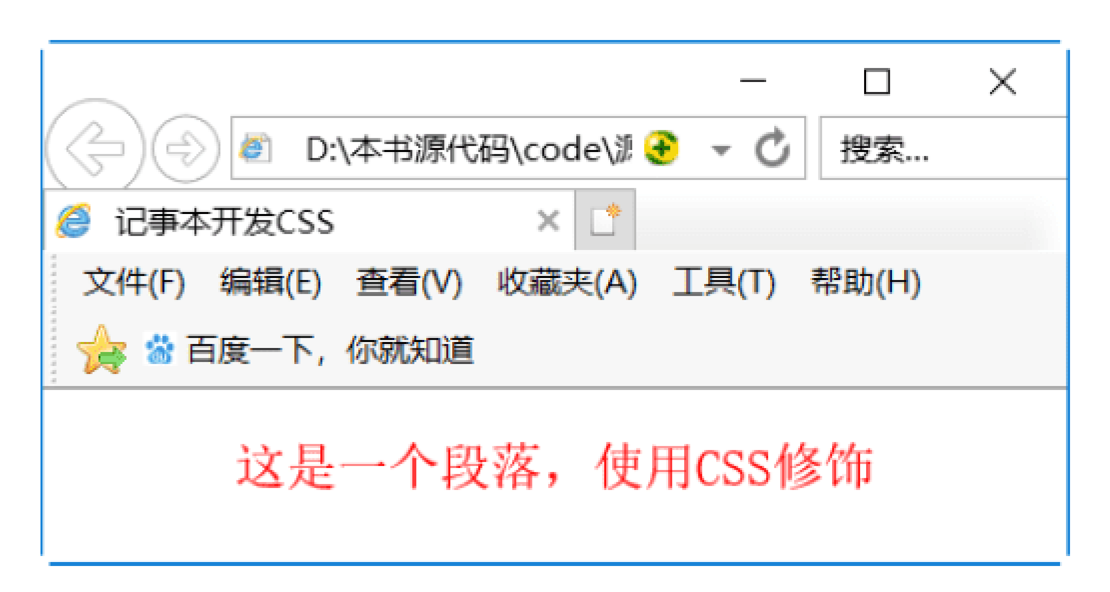


图 9-3　CSS 样式显示窗口

9.2.3　使用 Dreamweaver CC 创建 CSS 文件

随着 Web 技术的发展，越来越多的开发人员开始使用功能更多、界面更友好的专业 CSS 编辑器（如 Dreamweaver CC 的 CSS 编辑器和 Visual Studio 的 CSS 编辑器），这些编辑器有语法着色，并带有输入提示，甚至有自动创建 CSS 的功能。

【例 9.2】使用 Dreamweaver CC 创建 CSS 文件。

01 创建 HTML 文档。

使用 Dreamweaver CC 创建 HTML 文档的方法前面已经介绍过，这里就不再赘述了。此处创建了一个名称为 9.2.html 文档，如图 9-4 所示。

02 添加 CSS 样式。

❶ 在设计模式中，选中“使用 CSS 标记修饰”段落后，右击并在弹出的快捷菜单中选择【CSS 样式】➢【新建】命令，弹出如图 9-5 所示的对话框。





图 9-4　网页显示窗口

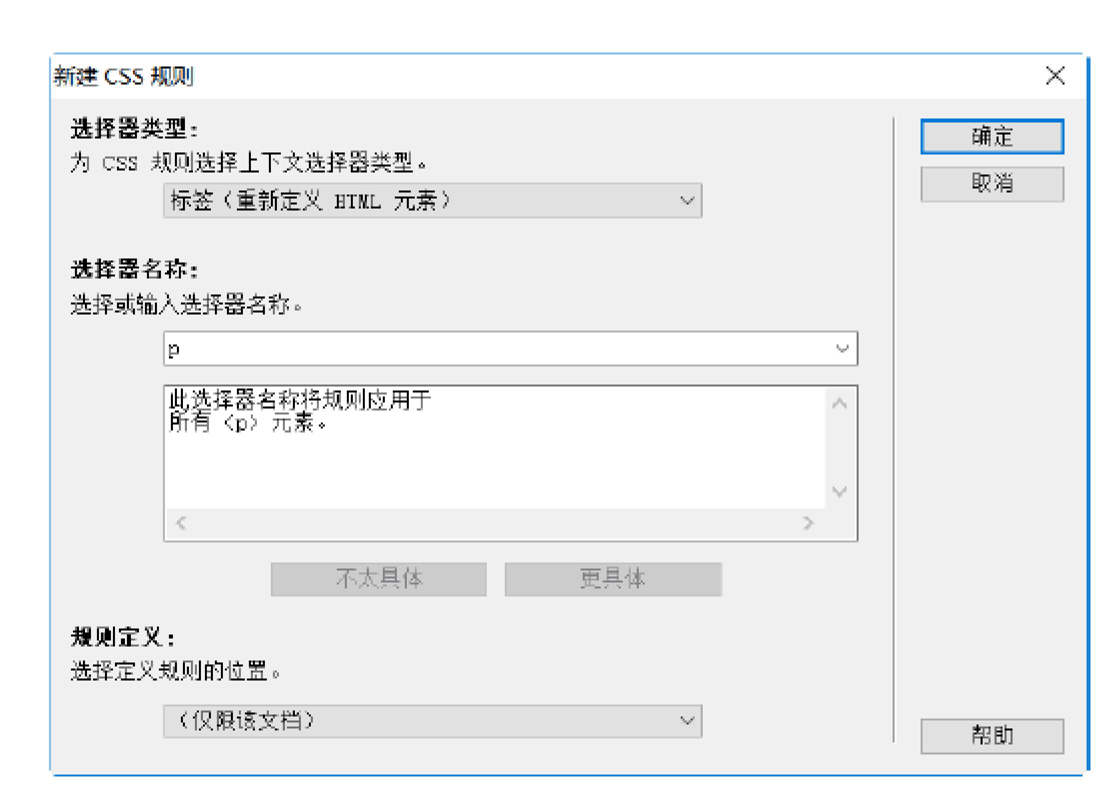


图 9-5　“新建 CSS 规则”对话框

❷在【为 CSS 规则选择上下文选择器类型】下拉列表中选择【标签（重新定义 HTML 元素）】选项（学习后面章节内容后，读者可以根据需要选择不同的选择器类型），然后单击【确

定】按钮，即可显示如图 9-6 所示的对话框。

❸在对话框中可以设置 p 的样式，设置完成后单击【确定】按钮。此时，HTML 文档内容发生了变化，如图 9-7 所示。

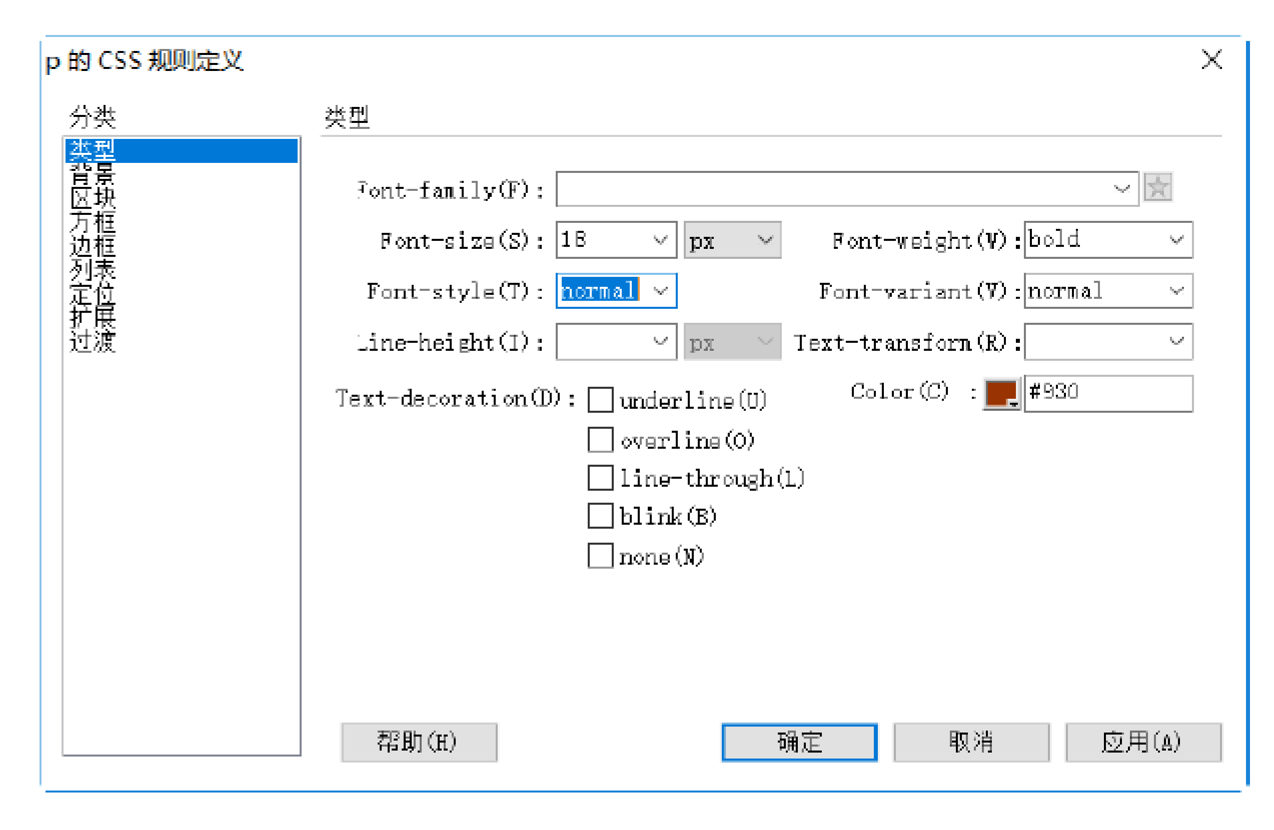


图 9-6 “p 的 CSS 规则定义”对话框

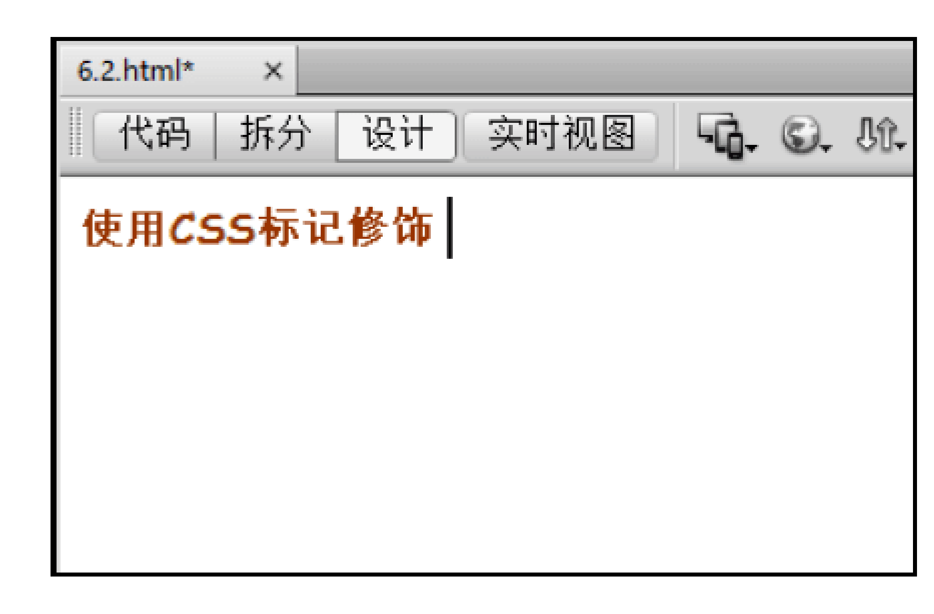


图 9-7 设置完成显示

从代码模式可以看到，在 head 部分中增加了一个<style>标记放置 CSS 样式，该样式用来修饰段落 p。

03 运行 HTML 文档

在 IE 11.0 中预览该网页，其显示结果如图 9-8 所示，可以看到字体颜色设置为浅红色，大小为 18 像素，字体加粗。

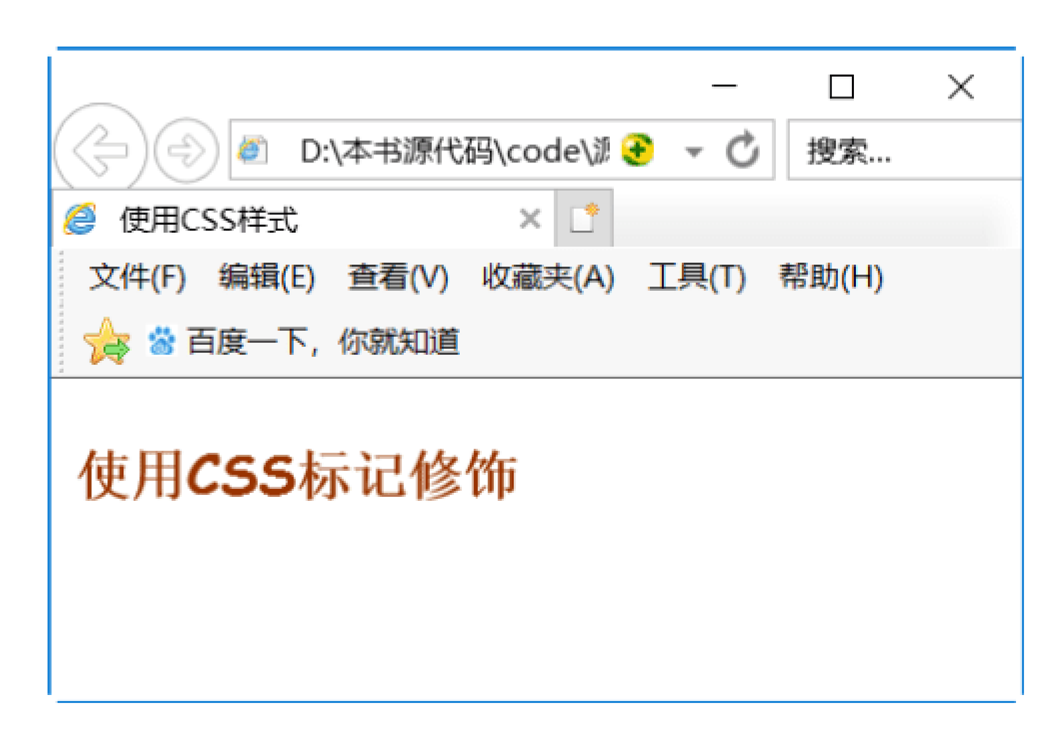


图 9-8 CSS 样式显示

上述使用 Dreamweaver 创建 CSS 的方法，只是其中一种，读者还可以直接在代码模式中编写 CSS 代码，此时会有很好的语法提示。

9.3 在 HTML5 中使用 CSS3 的方法

CSS 样式表能很好地控制页面显示，分离网页内容和样式代码，它控制 HTML5 页面效果

的方式通常包括行内样式、内嵌样式、链接样式和导入样式。

9.3.1 行内样式

行内样式是所有样式中比较简单、直观的方法，它可以直接把 CSS 代码添加到 HTML 文件中，是作为 HTML 的标记属性存在的。通过这种方法，可以很简单地对某个元素单独定义样式。

使用行内样式方法直接在 HTML 标记中使用 style 属性，该属性的内容就是 CSS 的属性和值。例如：

```
<p style="color:red">段落样式</p>
```

【例 9.3】（实例文件：ch09\9.3.html）

```
<!DOCTYPE html>
<html>
<head>
<title>行内样式</title>
</head>
<body>
<p
style="color:red;font-size:20px;text-decoration:underline;text-align:center
">此段落使用行内样式修饰</p>
<p style="color:blue;font-style:italic">正文内容</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-9 所示，可以看到两个 p 标记中都使用了 style 属性，并且设置了 CSS 样式，各个样式之间互不影响，分别显示自己的样式效果。第一个段落为红色字体，居中显示且带有下画线；第二个段落为蓝色字体，并以斜体显示。

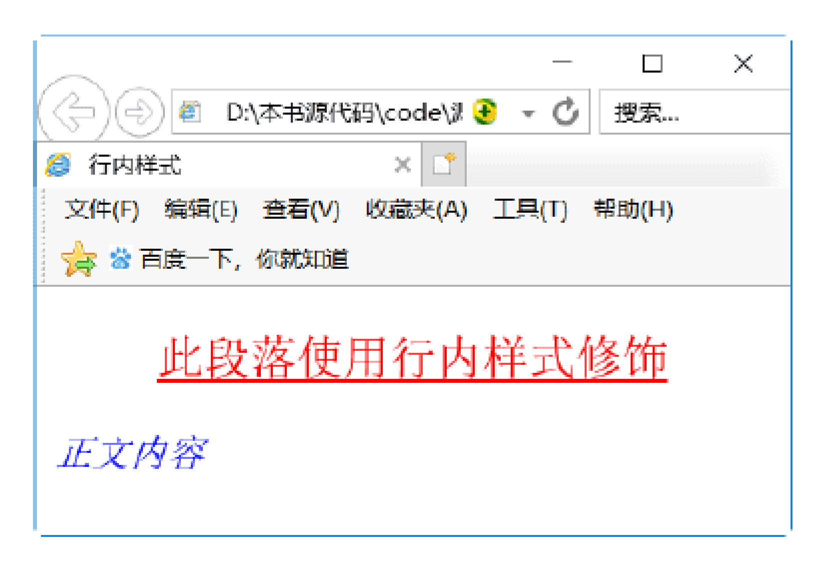


图 9-9 行内样式显示

尽管行内样式简单，但这种方法并不常用，因为这样添加无法完全发挥样式表“内容结构和样式控制代码分离”的优势，而且这种方式也不利于样式的重用。如果为每一个标记都设置 style 属性，那么后期维护成本会过高，网页也容易过胖，故不推荐使用。

9.3.2 内嵌样式

内嵌样式就是将 CSS 样式代码添加到<head>与</head>之间，并且用<style>和</style>标记进行声明。这种写法虽然没有实现页面内容和样式控制代码完全分离，但可以用于设置一些比较简单且需要样式统一的页面。其格式如下：

```
<head>
<style type="text/css">
p{color:red;font-size:12px;}
</style>
</head>
```

有些较低版本的浏览器不识别<style>标记，不能将样式正确地应用到页面显示上，而是直接将标记中的内容以文本的形式显示。为了解决此类问题，可以使用 HMTL 注释将标记中的内容隐藏。如果浏览器能够识别<style>标记，则标记内被注释的 CSS 样式定义代码依旧能够发挥作用。

```
<head>
<style type="text/css">
<!—
  p{color:red;font-size:12px;}
  -->
</style>
</head>
```

【例 9.4】（实例文件：ch09\9.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>内嵌样式</title>
<style type="text/css">
p{
   color:orange;          /*设置字体颜色为橙色*/
   text-align:center;     /*设置段落居中显示*/
   font-weight:bolder;    /*设置字体加粗效果*/
   font-size:25px;       /*设置字体大小*/
}
</style>
</head><body>
<p>此段落使用内嵌样式修饰</p>
<p>正文内容</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-10 所示，可以看到两个段落都被 CSS 样式修饰且样式保持一

致，均为段落居中、加粗并以橙色字体显示。

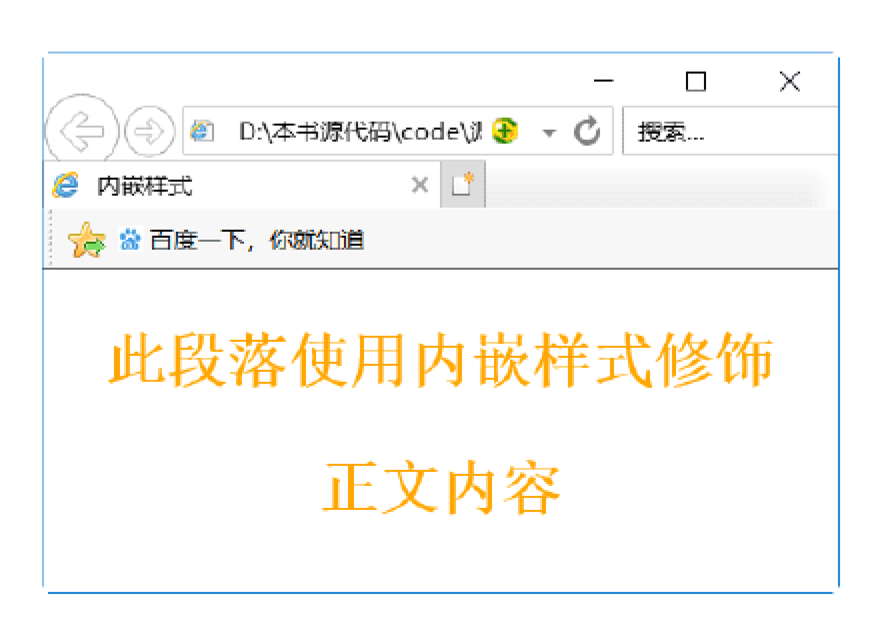


图 9-10　内嵌样式显示

在上面的例子中，所有 CSS 编码都在 style 标记里，既方便了后期维护，页面与行内样式相比较也大大瘦身了。但如果一个网站拥有很多页面，且对于不同页面段落都希望采用同样的风格时，内嵌方式就显得有点麻烦。该方法只适用于特殊页面设置单独的样式风格。

9.3.3　链接样式

链接样式是 CSS 中使用频率最高，也是最实用的方法。它可以很好地将“页面内容”和“样式风格代码”分离成两个文件或多个文件，实现了页面框架 HTML 代码和 CSS 代码的完成分离，使前期制作和后期维护都十分方便。同一个 CSS 文件，根据需要可以链接到网站中所有的 HTML 页面上，使得网站整体风格统一，并且后期维护的工作量也大大减少。

链接样式是指在外部定义 CSS 样式表并形成以.css 为扩展名的文件，然后在页面中通过<link>标记链接到页面中。该链接语句必须放在页面的<head>标记区，代码如下：

```
<link rel="stylesheet" type="text/css" href="1.css" />
```

（1）rel 表示链接到样式表，其值为 stylesheet。

（2）type 表示样式表类型为 CSS 样式表。

（3）href 指定了 CSS 样式表文件的路径，此处表示当前路径下名称为 1.css 文件。

这里使用的是相对路径。如果 HTML 文档与 CSS 样式表没有在同一路径下，则需要指定样式表的绝对路径或引用位置。

【例 9.5】（实例文件：ch09\9.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>链接样式</title>
<link rel="stylesheet" type="text/css" href="9.5.css"/>
</head>
```

```
<body>
<h1>CSS学习</h1>
<p>此段落使用链接样式修饰</p>
</body>
</html>
```

【例 9.5】（实例文件：ch09\9.5.css）

```
h1{
text-align:center;  /*设置标题居中显示*/
}
p{
font-weight:29px;   /*设置字体的粗细*/
text-align:center;  /*设置段落居中显示*/
font-style:italic;  /*设置字体样式为斜体*/
}
```

在 IE 11.0 中浏览效果如图 9-11 所示，其中标题和段落以不同样式显示，标题居中显示，段落以斜体居中显示。



图 9-11　链接样式显示

链接样式最大的优势就是将 CSS 代码和 HTML 代码完全分离，并且同一个 CSS 文件能被不同的 HTML 文件链接使用。



在设计整个网站时，为了实现相同的样式风格，可以将同一个 CSS 文件链接到所有的页面中。如果整个网站需要修改样式，则只修改 CSS 文件即可。

9.3.4　导入样式

导入样式和链接样式基本相同，都需要创建一个单独的 CSS 文件，然后将其引入到 HTML 文件中，只不过语法和运作方式有所差别。采用导入样式是在 HTML 文件初始化时，会被导入到 HTML 文件内作为文件的一部分，类似于内嵌效果。而链接样式则是在 HTML 标记需要样式风格时才以链接方式引入。

导入外部样式表是指在内嵌样式表的<style>标记中使用@import 导入一个外部的 CSS 文件。例如：

```
<head>
<style type="text/css">
<!--
  @import "1.css"
-->
</style>
</head>
```

导入外部样式表相当于将样式表导入到内嵌样式表中，其中@import 必须在样式表的开始部分（即位于其他样式表代码的上面）。

【例 9.6】（实例文件：ch09\9.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>导入样式</title>
<style>
@import "9.6.css"
</style>
</head>
<body>
<h1>CSS学习</h1>
<p>此段落使用导入样式修饰</p>
</body>
</html>
```

【例 9.6】（实例文件：ch09\9.6.css）

```
h1{text-align:center;          /*设置标题居中显示*/
color:#0000ff;                 /*设置标题的颜色为蓝色*/
}
p{font-weight:bolder;          /*设置字体的粗细*/
text-decoration:underline;    /*设置下画线效果*/
font-size:20px;               /*设置字体的大小*/
}
```

IE 11.0 中浏览效果如图 9-12 所示，其中标题和段落以不同样式显示，标题居中显示、颜色为蓝色，段落的字体设置为加粗、下画线，大小为 20 像素。

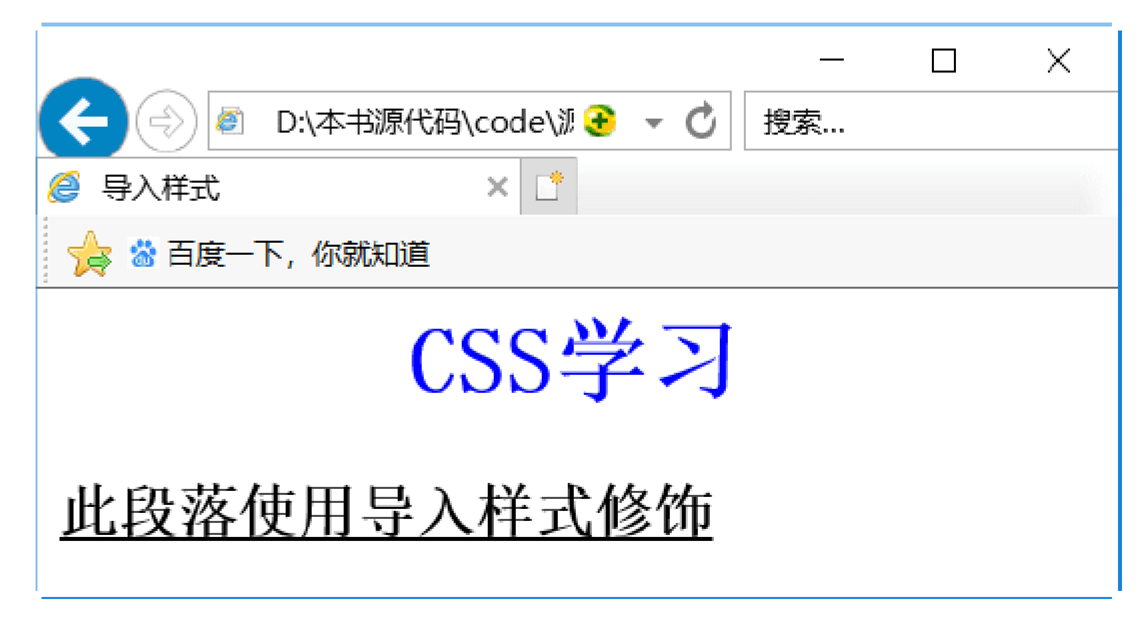


图 9-12　导入样式显示

导入样式与链接样式比较，最大的优势就是可以一次导入多个 CSS 文件。其格式如下：

```
<style>
@import "9.6.css"
@import "test.css"
</style>
```

9.3.5 优先级问题

如果同一个页面采用了多种 CSS 样式表方式（如同时使用行内样式、链接样式和内嵌样式），且这几种方式共同作用于同一属性，就会出现优先级问题。例如使用内嵌样式设置字体为宋体，使用链接样式设置字体为红色，那么二者会同时生效，但如果都设置字体颜色且颜色不同，那么哪种样式的设置才有效呢？

1. 行内样式和内嵌样式比较

例如，有这样一种情况：

```
<style>
p{color:red}
</style>
<p style="color:blue">段落应用样式</p>
```

在样式定义中，段落标记<p>匹配了两种样式规则，一种使用内样式定义颜色为红色，一种使用 p 行内样式定义颜色为蓝色，而在页面代码中，该标记使用了类选择符。但是，标记内容最终会以哪一种样式显示呢？

【例 9.7】（实例文件：ch09\9.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title>优先级比较</title>
<style>
p{color:red}
</style>
</head>
<body>
<p style="color:blue">优先级测试</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-13 所示，段落以蓝色字体显示，可以看出行内样式优先级大于内嵌样式。

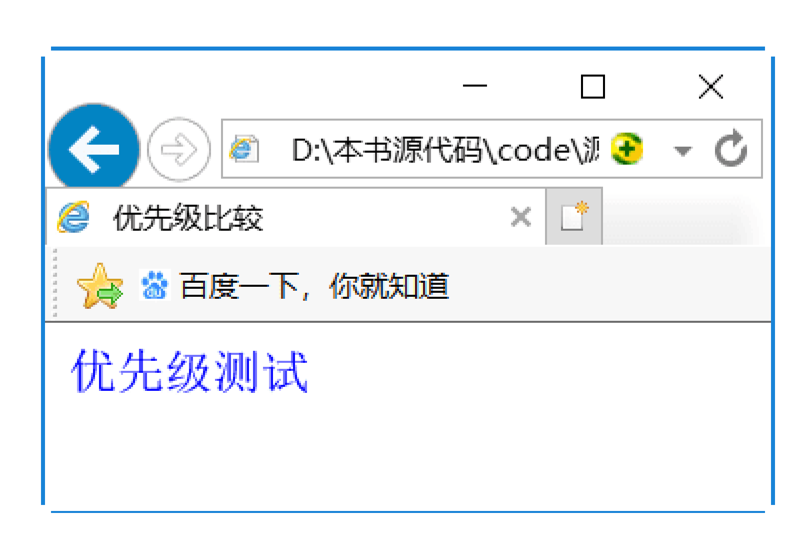


图 9-13　优先级显示

2. 内嵌样式和链接样式比较

以相同例子测试内嵌样式和链接样式的优先级。将相应的颜色样式代码单独放在一个 CSS 文件中，供链接样式引用。

【例 9.8】（实例文件：ch09\9.8.html）

```
<!DOCTYPE html>
<html>
<head>
<title>优先级比较</title>
<link href="9.8.css" type="text/css" rel="stylesheet">
<style>p{color:red}
</style></head>
<body>
<p>优先级测试</p>
</body>
</html>
```

【例 9.8】（实例文件：ch09\9.8.css）

```
p{color:yellow}
```

在 IE 11.0 中浏览效果如图 9-14 所示，段落以红色字体显示，可以看出内嵌样式优先级大于链接样式。

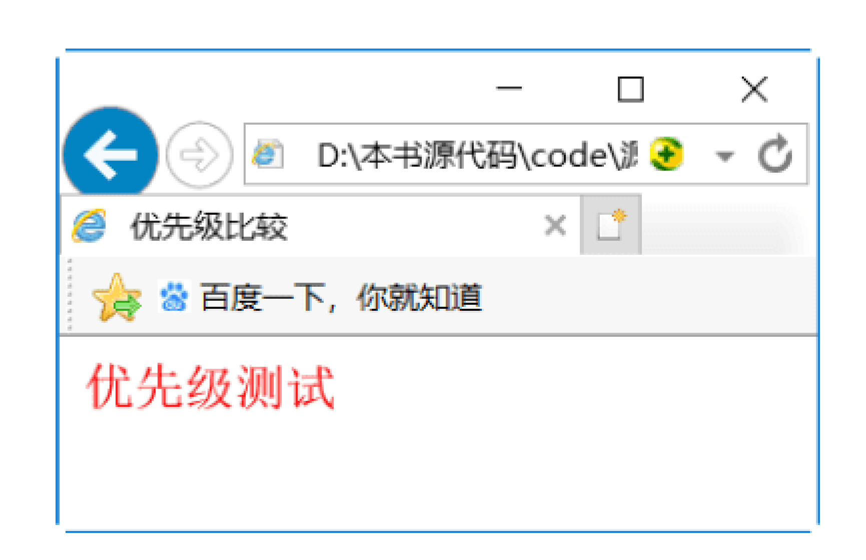


图 9-14　优先级测试

3. 链接样式和导入样式

现在进行链接样式和导入样式的优先级比较。分别创建两个 CSS 文件，一个作为链接，

一个作为导入。

【例 9.9】（实例文件：ch09\9.9.html）

```
<!DOCTYPE html>
<html>
<head>
<title>优先级比较</title>
<style>
@import "9.9 2.css"
</style>
<link href="9.9 1.css" type="text/css" rel="stylesheet">
</head>
<body>
<p>优先级测试</p>
</body>
</html>
```

【例 9.9】（实例文件：ch09\9.9_1.css）

```
p{color:green}
```

【例 9.9】（实例文件：ch09\9.9_2.css）

```
p{color:purple}
```

在 IE 11.0 中浏览效果如图 9-15 所示，段落以绿色显示，可以看出链接样式优先级大于导入样式。

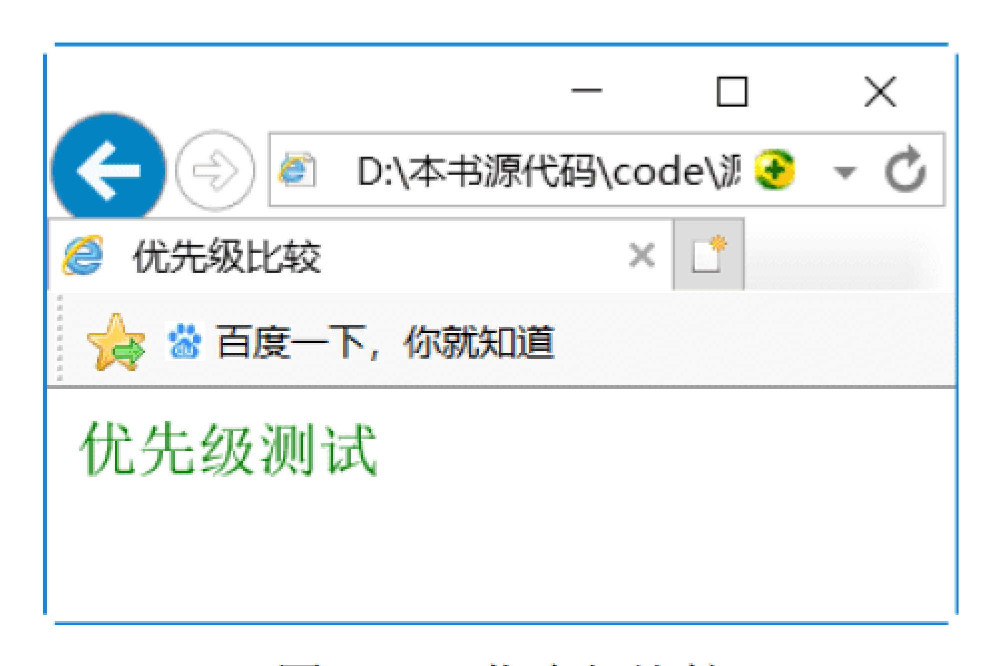


图 9-15　优先级比较

通过比较，CSS 样式表方式的优先级顺序由大到小依次为：行内样式、内嵌样式、链接样式和导入样式。

9.4　CSS3 选择器

选择器（selector）也被称为选择符。所有 HTML 语言中的标记都是通过不同的 CSS 选择器进行控制的。选择器不只是 HMTL 文档中的元素标记，它还可以是类（Class，这不同于面

向对象程序设计语言中的类）、ID（元素的唯一特殊名称，便于在脚本中使用）或是元素的某种状态（如 a:link）。根据 CSS 选择器用途可以把选择器分为标记选择器、类选择器、全局选择器、ID 选择器和伪类选择器等。

9.4.1 标记选择器

HTML 文档是由多个不同标记组成的，而 CSS 选择器就是声明那些标记的样式风格。例如，p 选择器就是用于声明页面中所有<p>标记的样式风格，同样也可以通过 h1 选择器来声明页面中所有<h1>标记的样式风格。

标记选择器基本的形式如下：

```
tagName{property:value}
```



其中 tagName 表示标记名称，如 p、h1 等 HTML 标记；property 表示 CSS3 属性；value 表示 CSS3 属性值。

通过声明一个具体标记，可以对文档里这个标记出现的每一个地方应用样式定义，这种做法通常用在设置整个网站都会出现的基本样式。例如，下面的定义就用于为一个网站设置默认字体。

```
body, p, td, th, div, blockquote, dl, ul, ol {
  font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif;
  font-size: 1em;
  color: #000000;
}
```

这个选择器声明了一系列的标记，所有这些标记出现的地方都将以定义的样式（字体、字号和颜色）显示。理论上仅声明〈body〉标记就已经足够（因为所有其他标记会出现在〈body〉标记内部，并且将因此继承它的属性），但是许多浏览器不能恰当地将这些样式属性带入表格和其他标记里。因此，为了避免这种情况这里声明了其他标记。

【例 9.10】（实例文件：ch09\9.10.html）

```
<!DOCTYPE html>
<html>
<head>
<title>标记选择器</title>
<style>
p{color:blue;        /*设置字体的颜色为蓝色*/
font-size:20px;   /*设置字体的大小*/
}
</style>
</head>
```

```
<body>
<p>此处使用标记选择器控制段落样式</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-16 所示，可以看到段落字体以蓝色显示，大小为 20 像素。

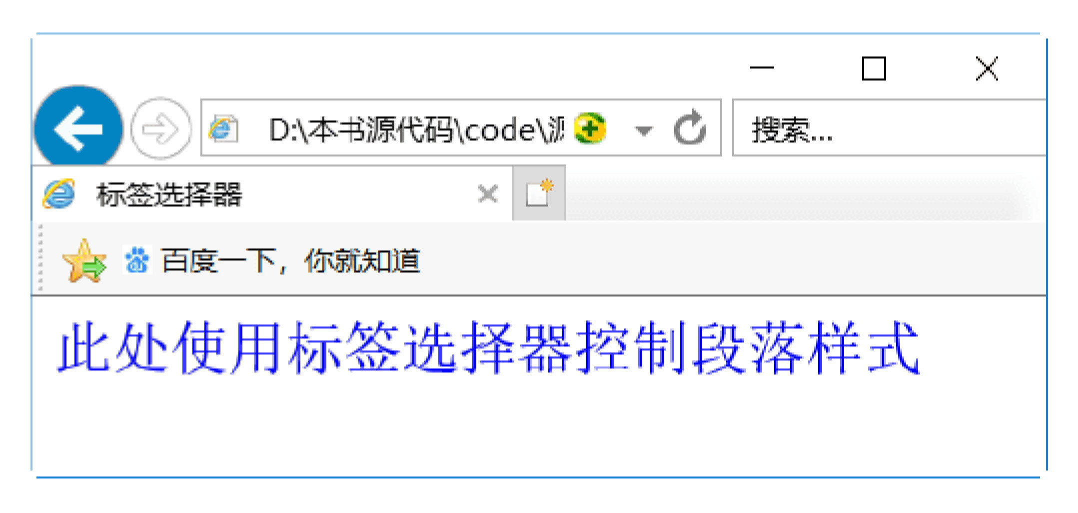


图 9-16 标记选择器显示

如果在后期维护中需要调整段落颜色，则只修改 color 属性值即可。



CSS3 标准对于所有属性和值都有相对严格的要求，如果声明的属性在 CSS3 规范中没有或某个属性值不符合属性要求，都不能使 CSS 语句生效。

9.4.2 类选择器

使用标记选择器可以控制该页面中所有相关标记的显示样式，如果需要对其中一系列标记重新设置，此时仅使用标记选择器是远远不够的，还需要使用类选择器。

类选择器用来为一系列标记定义相同的呈现方式，常用的语法格式如下：

```
.classValue{property:value}
```

classValue 是选择器的名称，具体名称由 CSS 制定者自己命名。如果一个标记具有 class 属性且 class 属性值为 classValue，那么该标记的呈现样式由该选择器指定。在定义类选择符时，需要在 classValue 前面加一个句点“.”，例如：

```
.rd{color:red}            /*设置字体的颜色为红色*/
.se{font-size:3px}        /*设置字体的大小*/
```

上面定义了两个类选择器，分别为 rd 和 se。类的名称可以是任意英文字符串或以英文开头与数字的组合，一般情况下采用其功能或效果的缩写。

在<p>标记的 class 属性中使用类选择符。

<p class="rd">class 属性是被用来引用类选择器的属性</p>。

类选择器只能被应用于指定的标记中（如<p>标记），可以在不同标记中使用相同的呈现方式，如下所示。

```
<p class="rd">段落样式</p>
<h3 class="rd">标题样式</h3>
```

【例 9.11】（实例文件：ch09\9.11.html）

```
<!DOCTYPE html>
<html>
<head><title>类选择器</title>
<style>
.aa{
   color:blue;      /*设置字体的颜色为蓝色*/
   font-size:20px; /*设置字体的大小为20px*/
}
.bb{
  color:red;       /*设置字体的颜色为红色*/
  font-size:22px; /*设置字体的大小为22px*/
}
</style></head><body>
<h3 class="bb">学习类选择器</h3>
<p class="aa">此处使用类选择器aa控制段落样式</p>
<p class="bb">此处使用类选择器bb控制段落样式</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-17 所示，可以看到第一段落字体以蓝色显示，大小为 20 像素；第二段落字体以红色显示，大小为 22 像素；标题字体同样以红色显示，大小为 22 像素。

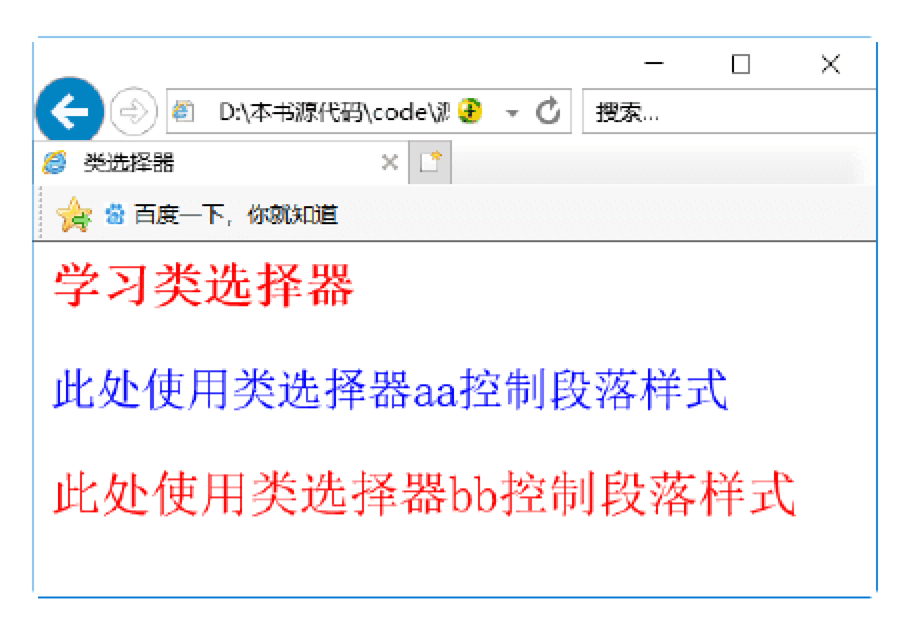


图 9-17　类选择器显示

9.4.3　ID 选择器

ID 选择器和类选择器类似，都是针对特定属性的属性值进行匹配的。ID 选择器定义的是某一个特定的 HTML 标记，一个网页文件中只能有一个标记使用某一 ID 的属性值。

定义 ID 选择器的基本语法格式如下：

```
#idValue{property:value}
```

在上述基本语法格式中，idValue 是选择器名称，可以由 CSS 定义者自己命名。如果某标

记具有 id 属性，并且该属性值为 idValue，那么该标记的呈现样式由该 ID 选择器指定。在正常情况下，id 属性值在文档中具有唯一性。在定义 ID 选择器时，需要在 idValue 前面加一个“#”符号，例如：

```
#fontstyle
{
  color:red;              /*设置字体的颜色为红色*/
  font-weight:bold;      /*设置字体的粗细*/
  font-size:large       /*设置字体的大小*/
}
```

与类选择器相比，使用 ID 选择器定义样式是有一定局限性的。类选择器与 ID 选择器主要有以下两种区别：

（1）类选择器可以给任意数量的标记定义样式，但 ID 选择器在页面的标记中只能使用一次。

（2）ID 选择器比类选择器具有更高的优先级，即当 ID 选择器与类选择器发生冲突时，优先使用 ID 选择器定义的样式。

【例 9.12】（实例文件：ch09\9.12.html）

```
<!DOCTYPE html>
<html>
<head>
<title>ID选择器</title>
<style>
#fontstyle{
   color:blue;              /*设置字体的颜色为蓝色*/
   font-weight:bold;      /*设置字体的粗细*/
}
#textstyle{
  color:red;              /*设置字体的颜色为红色*/
  font-size:22px;        /*设置字体的大小*/
}
</style>
</head>
<body>
<h3 id="textstyle">学习ID选择器</h3>
<p id="textstyle">此处使用ID选择器textsyle控制段落样式</p>
<p id="fontstyle">此处使用ID选择器fontstyle控制段落样式</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-18 所示，可以看到第一段落字体以红色显示，大小为 22 像素；第二段落字体以蓝色显示，字体加粗；标题字体同样以红色显示，大小为 22 像素。

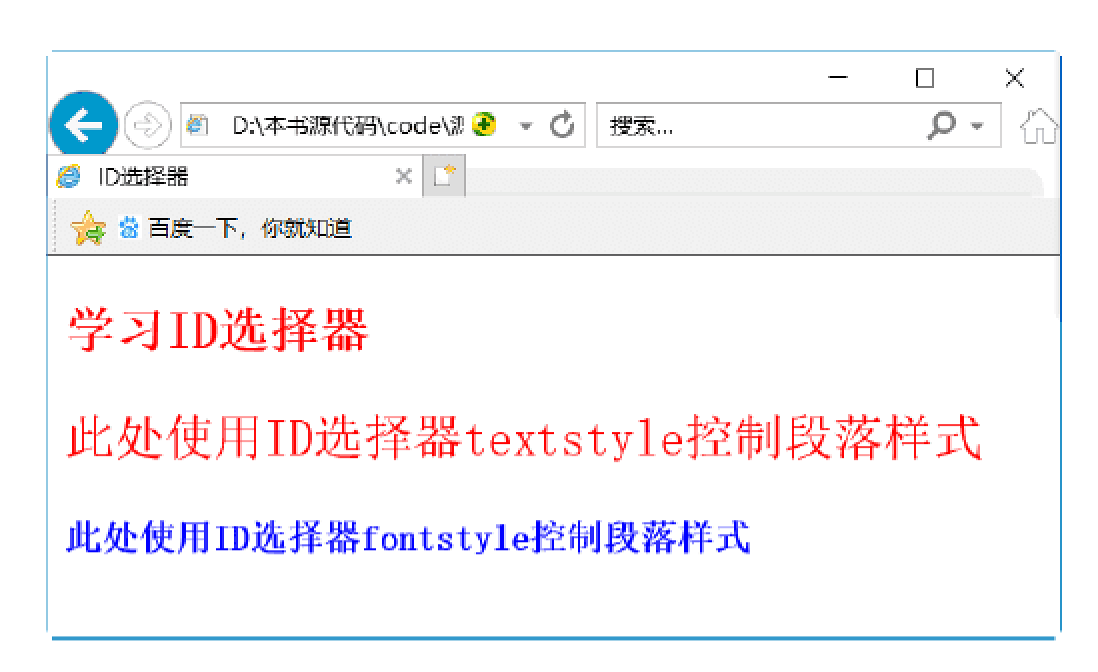


图 9-18　ID 选择器显示

从上面代码中可以看出，标题 h3 和第一段落都使用了名称为 textstyle 的 ID 选择器并都显示了 CSS 方案，但这里需要指出的是，将 ID 选择器用于多个标记是错误的，因为每个标记定义的 ID 不只是 CSS 可以调用，JavaScript 等脚本语言同样也可以调用。如果一个 HTML 中有两个相同 id 的标记，那么将会导致 JavaScript 在查找 id 时出错。



由于 JavaScript 等脚本语言也能调用 HTML 中设置的 id，因此 ID 选择器一直被广泛使用。网页设计者在编写 HTML 代码时应该养成一个习惯，一个 id 只赋予一个 HTML 标记。

9.4.4　全局选择器

如果想要一个页面中所有 HTML 标记使用同一种样式，就可以使用全局选择器。全局选择器，顾名思义就是对所有 HTML 标记起作用。其语法格式如下：

```
*{property:value}
```

其中，“*”表示对所有标记起作用；property 表示 CSS3 属性名称；value 表示属性值。示例如下：

```
*{margin:0; padding:0;}        /*设置所有标记的外边距和内边距都为0*/
```

【例 9.13】（实例文件：ch09\9.13.html）

```
<!DOCTYPE html>
<html>
<head><title>全局选择器</title>
<style>
*{
  color:red;          /*设置字体的颜色为红色*/
  font-size:30px     /*设置字体的大小为30px*/
}
</style>
</head>
```

```
<body>
<p>使用全局选择器修饰</p>
<p>第一段</p>
<h1>第一段标题</h1>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-19 所示，两个段落和标题都是以红色字体显示，字体大小为 30 像素。

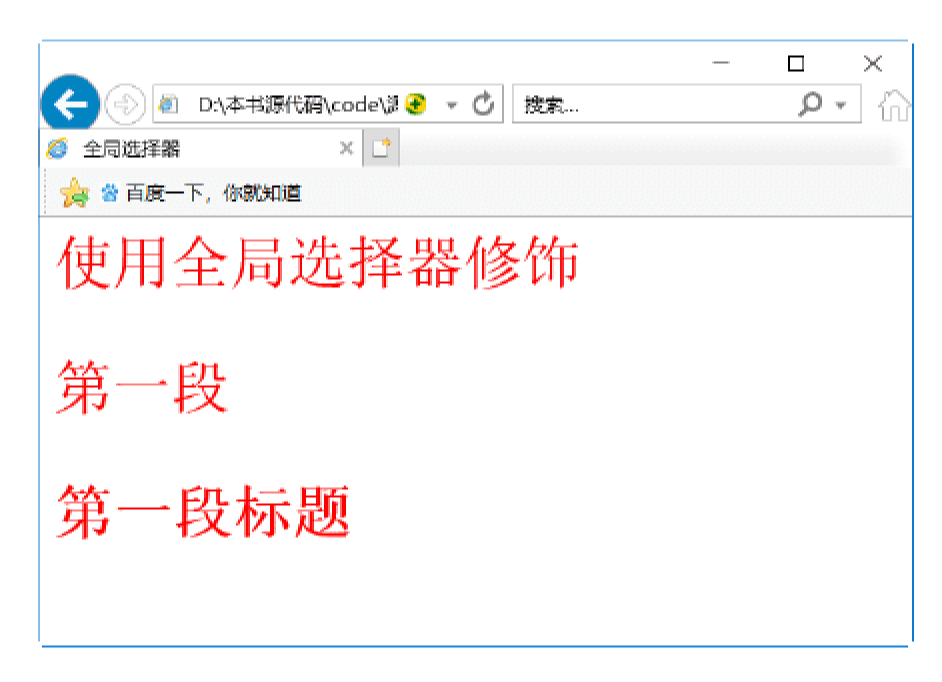


图 9-19　全局选择器

9.4.5　组合选择器

将多种选择器进行搭配，可以构成一种复合选择器，也称为组合选择器，即将标记选择器、类选择器和 ID 选择器组合起来使用。一般的组合方式是标记选择器和类选择器组合或标记选择器和 ID 选择器组合，由于这两种组合方式的原理和效果一样，所以本小节只介绍标记选择器和类选择器的组合。

组合选择器只是一种组合形式，并不是一种真正的选择器，但在实际应用中会经常使用。其语法格式如下：

```
tagName. class Value{property:value}
```

一般用在重复出现且样式相同的标记里，如 li 列表、td 单元格、和 dd 自定义列表等。例如：

```
h1.red{color: red}
<h1 class="red"></h1>
```

【例 9.14】（实例文件：ch09\9.14.html）

```
<!DOCTYPE html>
<html>
<head>
<title>组合选择器</title>
<style>
p{ /*标记选择器*/
```

```
    color:red
  }
  p.firstPar{/*组合选择器*/
    color:blue
  }
  .firstPar{/*类选择器*/
    color:green
  }
  </style>
  </head>
  <body>
  <p>这是普通段落</p>
  <p class="firstPar">此处使用组合选择器</p>
  <h1 class="firstPar">我是一个标题</h1>
  </body>
  </html>
```

在 IE 11.0 中浏览效果如图 9-20 所示，可以看到第一段落颜色为红色，采用的是标记选择器；第二段落显示的是蓝色，采用的是标记选择器和类选择器组合的选择器；标题以绿色字体显示，采用的是类选择器。

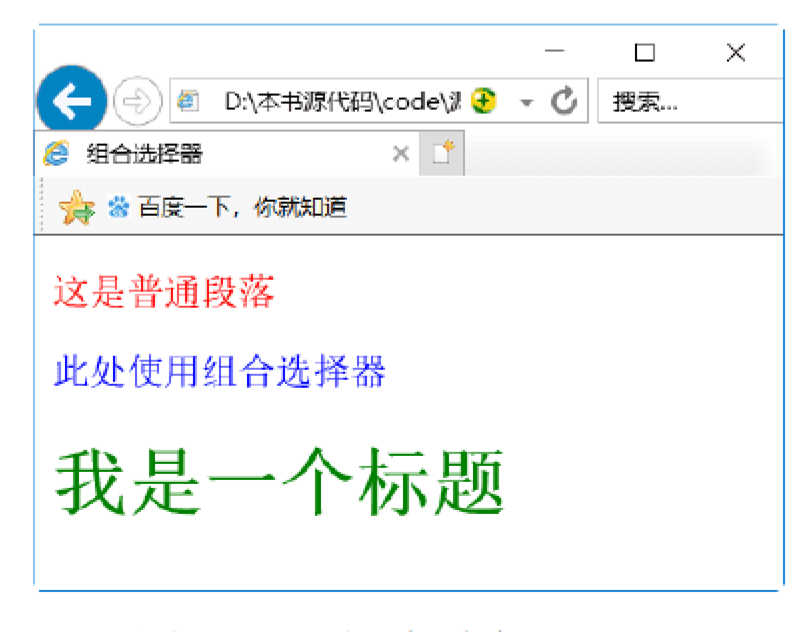


图 9-20　组合选择器显示

9.4.6　继承选择器

继承选择器的规则是：子标记在没有定义的情况下所有的样式是继承父标记的；当子标记重新定义父标记已经定义过的声明时，子标记会执行后面的声明，其中与父标记不冲突的地方仍然沿用父标记的声明。

使用继承选择器就必须先了解 HTML 文档树和 CSS 继承，这样才能够很好地运用继承选择器。每个 HTML 都可以被看作一个文档树，文档树的根部就是<html>标记，而<head>和<body>标记就是其标记。在<head>和<body>里的其他标记就是<html>标记的孙子标记，整个 HTML 就呈现一种祖先和子孙的树状关系。CSS 的继承是指子孙标记继承祖先标记的某些属性，示例如下：

```
<div class="test">
<span><img src="xxx" alt="示例图片"/></span>
</div>
```

对于上面代码而言，如果其修饰样式为下面代码：

```
.test span img {border:1px blue solid;}
```

则表示该选择器先找到 class 为 test 的标记，然后从它的子标记里查找<span>标记，再从<span>的子标记中找到<img>标记。也可以采用下面的形式：

```
div span img {border:1px blue solid;}
```

可以看出其规律是从左往右依次细化，最后锁定要控制的标记。

【例 9.15】（实例文件：ch09\9.15.html）

```
<!DOCTYPE html>
<html>
<head>
<title>继承选择器</title>
<style type="text/css">
h1{color:red; text-decoration:underline;}
h1 strong{color:#004400; font-size:40px;}
h1 font{font-size:20px;}
</style>
</head>
<body>
<h1>测试CSS的<strong>继承</strong>效果</h1>
<h1>此处使用继承<font>选择器</font>了么？</h1>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-21 所示，可以看到第一个标题颜色为红色，但是“继承”两个字使用绿色显示并且大小为 40 像素，除了这两个设置外的其他 CSS 样式都是继承父标记<h1>的样式（例如下画线设置）。第二个标题虽然使用了 font 标记修饰选择器，但其样式都是继承于父类标记<h1>。

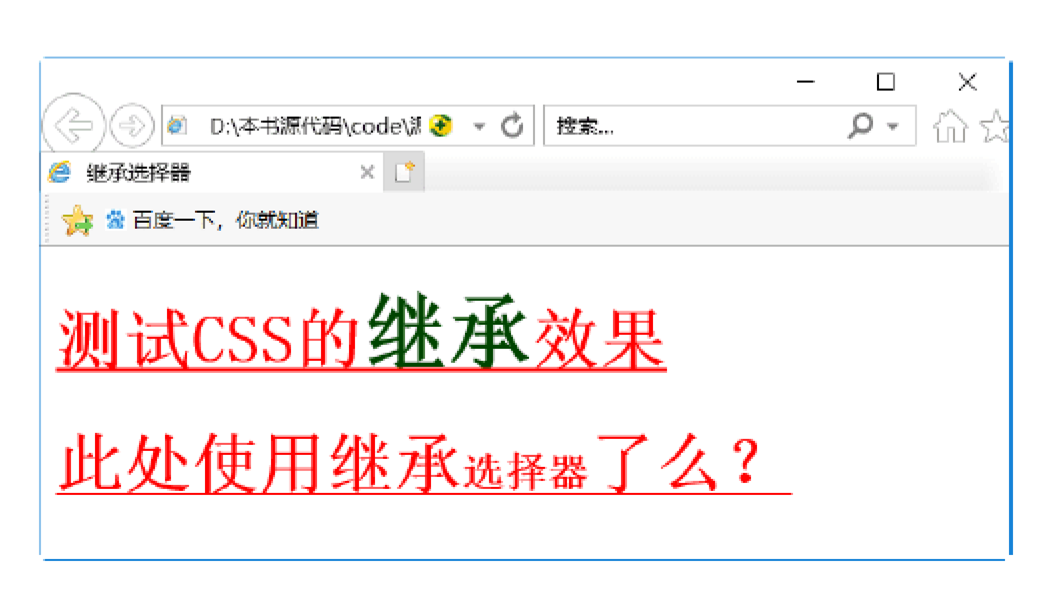


图 9-21　继承选择器

9.4.7 伪类

伪类也是选择器的一种，但是用伪类定义的 CSS 样式并不是作用在标记上的，而是作用在标记的状态上。由于很多浏览器支持不同类型的伪类，并且没有一个统一的标准，所以很多伪类都不常被用到。伪类包括:first-child、:link、:vistited、:hover、:active、:focus 和:lang 等。其中有一组伪类是主流浏览器都支持的，就是超链接的伪类，包括:link、:vistited、:hover 和:active。

伪类选择器定义的样式经常应用在标记<a>上，它表示超链接 4 种不同的状态，即未访问超链接（link）、已访问超链接（visited）、鼠标停留在超链接上（hover）和激活超链接（active）。需要注意的是，<a>标记可以只具有一种状态（:link），也可以同时具有两种或三种状态。比如说，任何一个有 href 属性的<a>标记，在未有任何操作时都已经具备了:link 的状态，也就是满足了有链接属性这个条件；如果访问过的<a>标记，就同时会具备 :link、visited 两种状态；把鼠标指针移到访问过的<a>标记上的时候，<a>标记就同时具备了 :link、visited、hover 三种状态。示例如下：

```
a:link{color:#FF0000; text-decoration:none}
a:visited{color:#00FF00; text-decoration:none}
a:hover{color:#0000FF; text-decoration:underline}
a:active{color:#FF00FF; text-decoration:underline}
```



上面的样式表示该超链接未访问时颜色为红色且无下画线，访问后是绿色且无下画线，鼠标指针放在超链接上为蓝色且有下画线，激活超链接时为紫色且有下画线。

【例 9.16】（实例文件：ch09\9.16.html）

```
<!DOCTYPE html>
<html>
<head>
<title>伪类</title>
<style>
a:link {color: red}      /* 未访问的链接 */
a:visited {color: green}    /* 已访问的链接 */
a:hover {color:blue}     /* 鼠标移动到链接上 */
a:active {color: orange}    /* 选定的链接 */
</style>
</head>
<body>
<a href="">链接到本页</a>
<a href="http://www.sohu.com">搜狐</a>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-22 所示，将鼠标指针停留在第一个超链接上时，显示颜色为蓝色；另一个是访问过后的超链接，显示颜色为绿色。

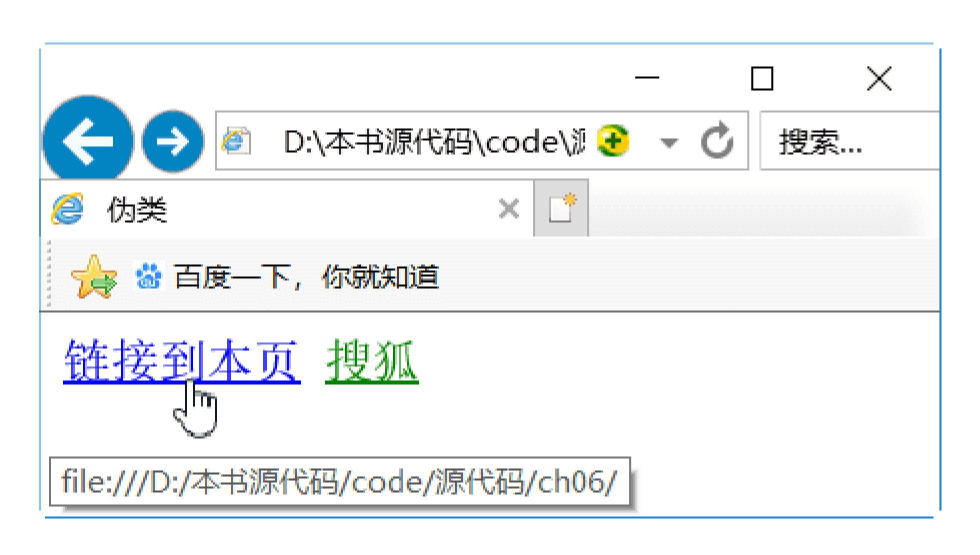


图 9-22 伪类显示

9.4.8 属性选择器

前面在使用 CSS3 样式对 HTML 标记进行修饰时，都是通过 HTML 标记名称或自定义名称指向具体的 HTML 元素，进而控制 HTML 标记样式。那么能不能直接通过标记属性来进行修饰，而不通过标记名称或自定义名称呢？直接使用属性控制 HTML 标记样式的选择器称为属性选择器。

属性选择器根据某个属性是否存在属性值来寻找元素，因此能够实现某些非常有意思和强大的效果。CSS2 标准就已经出现了 4 个属性选择器，在 CSS3 标准中又新加了 3 个属性选择器，也就是说现在的 CSS3 标准共有 7 个属性选择器，它们共同构成了 CSS 功能强大的标记属性过滤体系。

在 CSS3 标准中，常见的属性选择器如表 9-1 所示。

表 9-1 常见属性选择器

属性选择器格式	说明
E[foo]	选择匹配 E 的元素，且该元素定义了 foo 属性。注意，E 选择器可以省略，表示选择定义了 foo 属性的任意类型元素
E[foo= "bar "]	选择匹配 E 的元素，且该元素将 foo 属性值定义为“bar”。注意，E 选择器可以省略，用法与上一个选择器类似
E[foo~= "bar "]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值是一个以空格符分隔的列表，其中一个列表的值为“bar”。注意，E 选择符可以省略，表示可以匹配任意类型的元素 例如，a[title~="b1"]匹配，而不匹配
E[foo|="en"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值是一个用连字符（-）分隔的列表，值开头的字符为"en"。 注意，E 选择符可以省略，表示可以匹配任意类型的元素。例如，[lang|="en"]匹配<body lang="en-us"></body>，而不是匹配<body lang="f-ag"></body>

（续表）

属性选择器格式	说明
E[foo^="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值包含了前缀为"bar"的子字符串。注意，E 选择符可以省略，表示可以匹配任意类型的元素。例如，body[lang^="en"]匹配<body lang="en-us"></body>，而不匹配<body lang="f-ag"></body>
E[foo$="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值包含后缀为"bar"的子字符串。注意 E 选择符可以省略，表示可以匹配任意类型的元素。例如，img[src$="jpg"]匹配，而不匹配
E[foo*="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值包含"b"的子字符串。注意，E 选择器可以省略，表示可以匹配任意类型的元素。例如，img[src$="jpg"]匹配，而不匹配

【例 9.17】（实例文件：ch09\9.17.html）

```
<!DOCTYPE html>
<html>
<head>
<title>属性选择器</title>
<style>
[align]{color:red}
[align="left"]{font-size:20px;font-weight:bolder;}
[lang^="en"]{color:blue;text-decoration:underline;}
[src$="gif"]{border-width:5px;boder-color:#ff9900}
</style>
</head>
<body>
<p align=center>这是使用属性定义样式</p>
<p align=left>这是使用属性值定义样式</p>
<p lang="en-us">此处使用属性值前缀定义样式</p>
<p>下面使用了属性值后缀定义样式
<img src="2.gif" border="1"/>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-23 所示，可以看到第一段落使用属性 align 定义样式，其字体颜色为红色；第二段落使用属性值 left 修饰样式，其字体颜色为红色，大小为 20 像素并加粗显示，是因为该段落使用了 align 属性；第三段落字体显示为蓝色，且带有下画线，是因为属性 lang 的值前缀为 en；最后一个图片以边框样式显示，是因为属性值后缀为 gif。



图 9-23　属性选择器显示

9.4.9　结构伪类选择器

结构伪类选择器（Structural pseudo-classes）是 CSS3 新增的类型选择器。顾名思义，结构伪类就是利用文档结构树（DOM）实现元素过滤。也就是说，通过文档结构的相互关系来匹配特定的元素，从而减少文档内对 class 属性和 id 属性的定义，使得文档更加简洁。

在 CSS3 版本中，新增结构伪类选择器如表 9-2 所示。

表 9-2　新增结构伪类选择器

选择器	含义
E:root	匹配文档的根元素，对于 HTML 文档，就是 HTML 元素
E:nth-child(n)	匹配其父元素的第 n 个子元素，第一个编号为 1
E:nth-last-child(n)	匹配其父元素的倒数第 n 个子元素，第一个编号为 1
E:nth-of-type(n)	与:nth-child()作用类似，但是仅匹配使用同种标签的元素
E:nth-last-of-type(n)	与:nth-last-child() 作用类似，但是仅匹配使用同种标签的元素
E:last-child	匹配父元素的最后一个子元素，等同于:nth-last-child(1)
E:first-of-type	匹配父元素下使用同种标签的第一个子元素，等同于:nth-of-type(1)
E:last-of-type	匹配父元素下使用同种标签的最后一个子元素，等同于:nth-last-of-type(1)
E:only-child	匹配父元素下仅有的一个子元素，等同于:first-child:last-child 或:nth-child(1):nth-last-child(1)
E:only-of-type	匹配父元素下使用同种标签的唯一一个子元素，等同于:first-of-type:last-of-type 或:nth-of-type(1):nth-last-of-type(1)
E:empty	匹配一个不包含任何子元素的元素。注意，文本节点也被看作子元素

【例 9.18】（实例文件：ch09\9.18.html）

```
<!DOCTYPE html>
<html>
<head><title>结构伪类</title>
<style>
tr:nth-child(even){
background-color:#96FED1
}
tr:last-child{font-size:20px;}
</style>
</head>
<body>
<table border=1 width=80%>
<th>姓名</th><th>编号</th><th>性别</th>
<tr><td>刘海松</td><td>006</td><td>男</td></tr>
<tr><td>王峰</td><td>001</td><td>女</td></tr>
<tr><td>李张力</td><td>002</td><td>男</td></tr>
<tr><td>于辉</td><td>008</td><td>男</td></tr>
<tr><td>张浩</td><td>004</td><td>女</td></tr>
<tr><td>刘永权</td><td>003</td><td>男</td></tr>
</table>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-24 所示，可以看到表格中奇数行显示指定颜色，并且最后一行字体的大小以 20 像素显示，其原因就是采用了结构伪类选择器。



姓名	编号	性别
刘海松	006	男
王峰	001	女
李张力	002	男
于辉	008	男
张浩	004	女
刘永权	003	男

图 9-24　结构伪类选择器

9.4.10　UI 元素状态伪类选择器

UI 元素状态伪类（The UI element states pseudo-classes）也是 CSS3 新增选择器，其中 UI 即 User Interface（用户界面）的简称。UI 设计则是指对软件的人机交互、操作逻辑、界面美观的整体设计。好的 UI 设计不仅是让软件变得有个性有品位，还要让软件的操作变得舒适、简单、自由，充分体现软件的定位和特点。

UI 元素的状态一般包括可用、不可用、选中、未选中、获取焦点、失去焦点、锁定、待

机等。CSS3 定义了 3 种常用的状态伪类选择器，详细说明如表 9-3 所示。

表 9-3　常用的状态伪类选择器

选择器	说明
E:enabled	选择匹配 E 的所有可用 UI 元素。注意，在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如 input:enabled 匹配<form><input type=text/><input type=buttion disabled=disabled/></form>代码中的文本框，而不匹配代码中的按钮
E:disabled	选择匹配 E 的所有不可用元素，注意，在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如 input:disabled 匹配<form><input type=text/><input type=buttion disabled=disabled/></form>代码中的按钮，而不匹配代码中的文本框
E:checked	选择匹配 E 的所有可用 UI 元素。注意，在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如 input:checked 匹配<form><input type=checkbox/><input type=radio checked=checked/></form>代码中的单选按钮，但不匹配该代码中的复选框

【例 9.19】（实例文件：ch09\9.19.html）

```
<!DOCTYPE html>
<html>
<head>
<title>UI元素状态伪类选择器</title>
<style>
input:enabled {border:1px dotted #666;background:#ff9900;}
input:disabled {border:1px dotted #999;background:#F2F2F2;}
</style>
</head>
<body>
<center>
<h3 align=center>用户登录</h3>
<form method="post" action="">
用户名：<input type=text name=name><br>
密  码：<input type=password name=pass disabled="disabled"><br>
<input type=submit value=提交>
<input type=reset value=重置>
</form>
<center>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-25 所示，可以看到表格中可用的表单元素都显示为浅黄色，而不可用的元素显示为灰色。

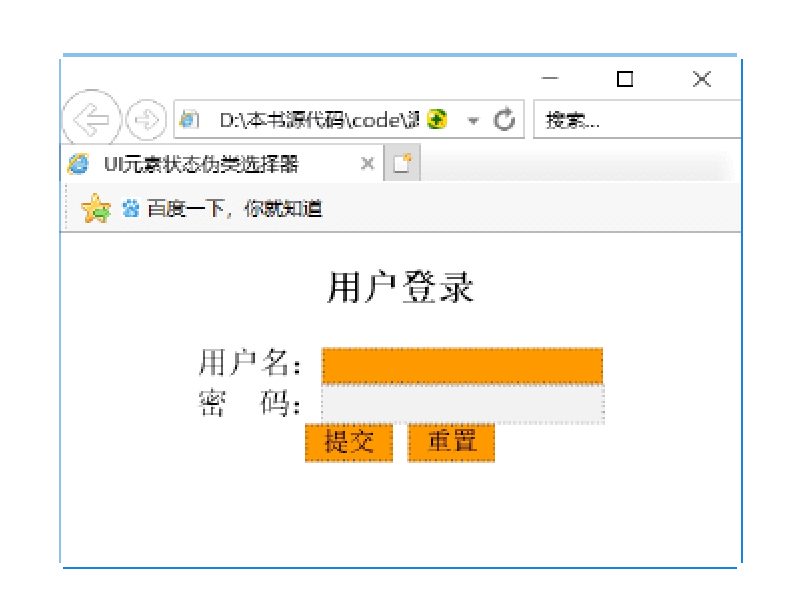


图 9-25　UI 元素状态伪类选择器应用

9.5　选择器声明

使用 CSS 选择器可以控制 HTML 标记样式，其中每个选择器属性可以一次声明多个，即创建多个 CSS 属性修饰 HTML 标记。实际上也可以将选择器声明多个，并且任何形式的选择器（如标记选择器、class 类别选择器、ID 选择器等）都是合法的。

9.5.1　集体声明

在一个页面中，有时需要不同种类的标记样式保持一致，例如需要<p>标记和<h1>标记的字体保持一致，此时可以将<p>标记和<h1>标记共同使用类选择器，除此之外还可以使用集体声明方法。集体声明就是在声明各种 CSS 选择器时，如果某些选择器的风格是完全相同的或者部分相同，可以将风格相同的 CSS 选择器同时声明。

【例 9.20】（实例文件：ch09\9.20.html）

```
<!DOCTYPE html>
<html>
<head>
<title>集体声明</title>
<style type="text/css">
h1,h2,p{
   color:red;             /*设置字体的颜色为红色*/
font-size:20px;           /*设置字体的大小*/
font-weight:bolder;       /*设置字体的粗细*/
}
</style>
</head>
<body>
<h1>此处使用集体声明</h1>
<h2>此处使用集体声明</h2>
<p>此处使用集体声明</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-26 所示，可以看到网页上标题 1、标题 2 和段落都以红色字体加粗显示，并且大小为 20 像素。

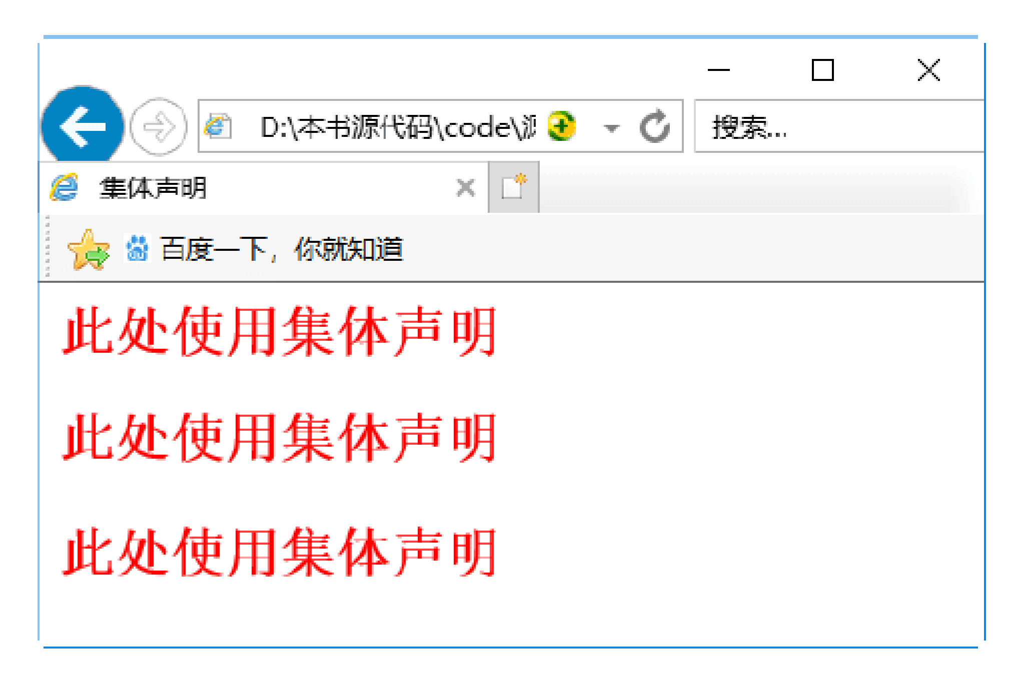


图 9-26 集体声明显示

9.5.2 多重嵌套声明

在 CSS 控制 HTML 标记样式时，还可以使用层层递进的方式（即嵌套方式）对指定位置的 HTML 标记进行修饰，例如当<p>与</p>之间包含<a>标记时，就可以使用这种方式对 HMTL 标记修饰。

【例 9.21】（实例文件：ch09\9.21.html）

```
<!DOCTYPE html>
<html>
<head>
<title>多重嵌套声明</title>
<style>
p{font-size:20px;}
p a{color:red;font-size:30px;font-weight:bolder;}
</style>
</head>
<body>
<p>这是一个多重嵌套<a href="">测试</a></p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-27 所示，可以看到在段落中超链接显示红色字体，大小为 30 像素，其原因是使用了嵌套声明。

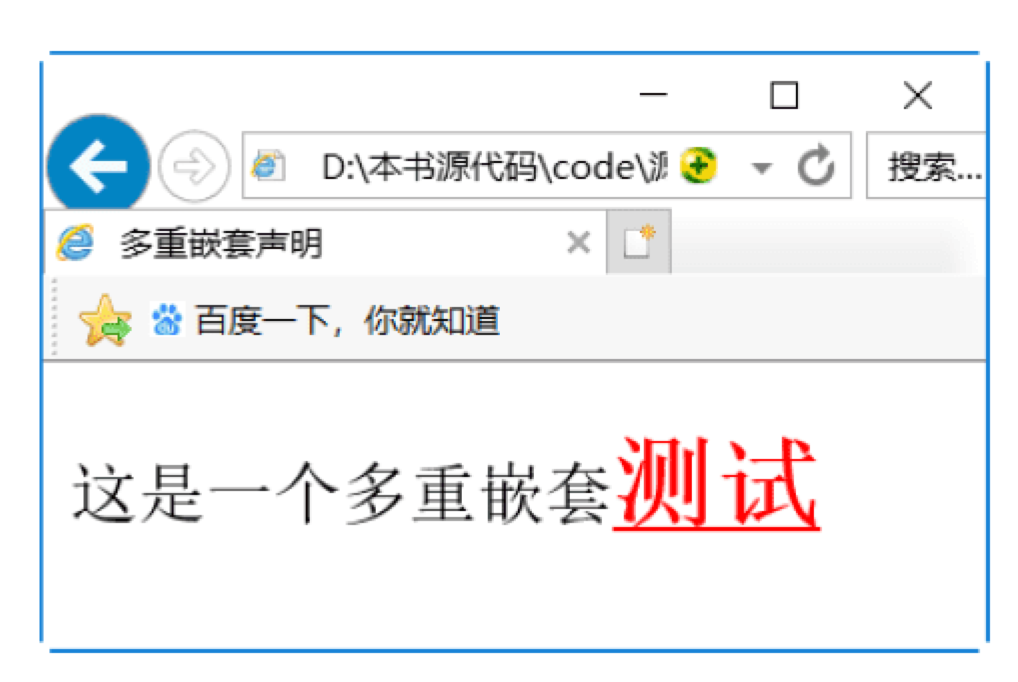


图 9-27　多重嵌套声明

9.6　综合实例 1——制作五彩标题

使用 CSS 可以给网页标题设置不同的字体样式，即建立一个 CSS 规则，将样式应用到页面中出现的所有<h1>标记或者是整个站点（当使用一个外部样式表的时候）。如果我们想改变整个站点上所有出现<h1>标记的颜色、尺寸、字体，则只需要修改一些 CSS 规则即可。

具体步骤如下：

01 分析需求。

本实例要求简单，首先使用标记<h1>创建一个标题，然后使用 CSS 样式对标题进行修饰，可以从颜色、尺寸、字体、背景、边框等方面入手。

02 构建 HTML 页面。

创建 HTML 页面，完成基本框架并创建标题。其代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>五彩标题</title>
</head>
<body>
<body>
<h1>
<span class=c1>美</span>
<span class=c2>食</span>
<span class=c3>介</span>
<span class=c4>绍</span></h1>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 9-28 所示，可以看到标题在网页中的显示没有任何修饰。

03 使用内嵌样式。

如果要对标题进行修饰，就需要添加 CSS（此处使用内嵌样式）。在<head>标记中添加 CSS，代码如下：

```
<style>
h1{}
</style>
```

此时没有任何变化，只是在代码中引入了<style>标记。

04 改变颜色、字体和尺寸。

添加 CSS 代码，改变标题样式。在颜色、字体和尺寸上面对其样式进行设置，代码如下：

```
h1{
font-family:Arial,sans-serif;    /*设置文本的字体样式*/
font-size:24px;                  /*设置字体的大小为24px*/
color:#369;                      /*设置字体颜色为浅蓝色*/
}
```

在 IE 11.0 中浏览效果如图 9-29 所示，可以看到字体大小为 24 像素，颜色为浅蓝色，字体为 Arial。

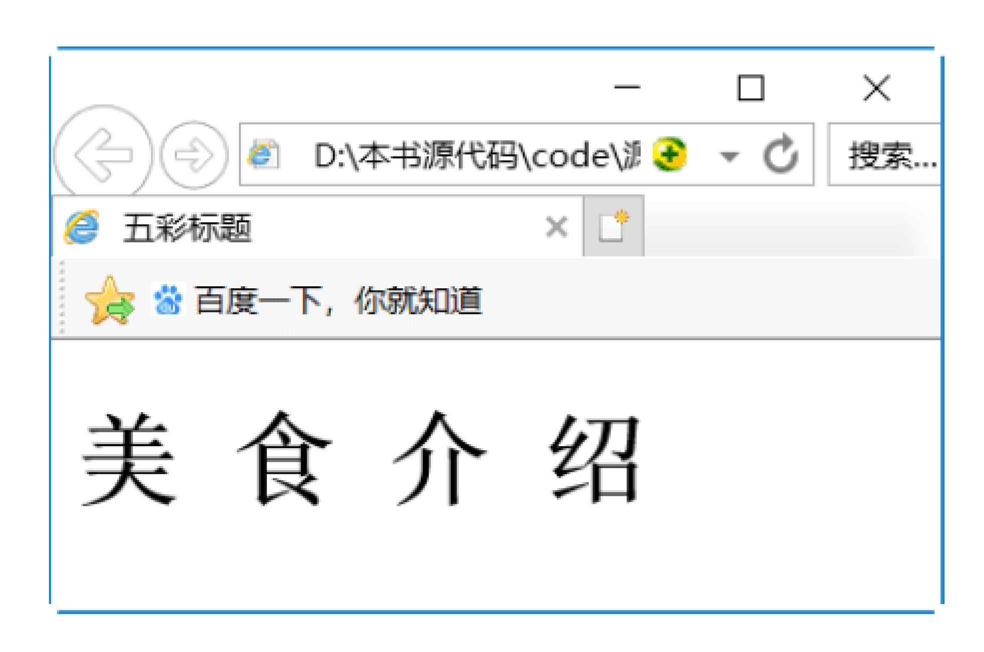


图 9-28　标题显示

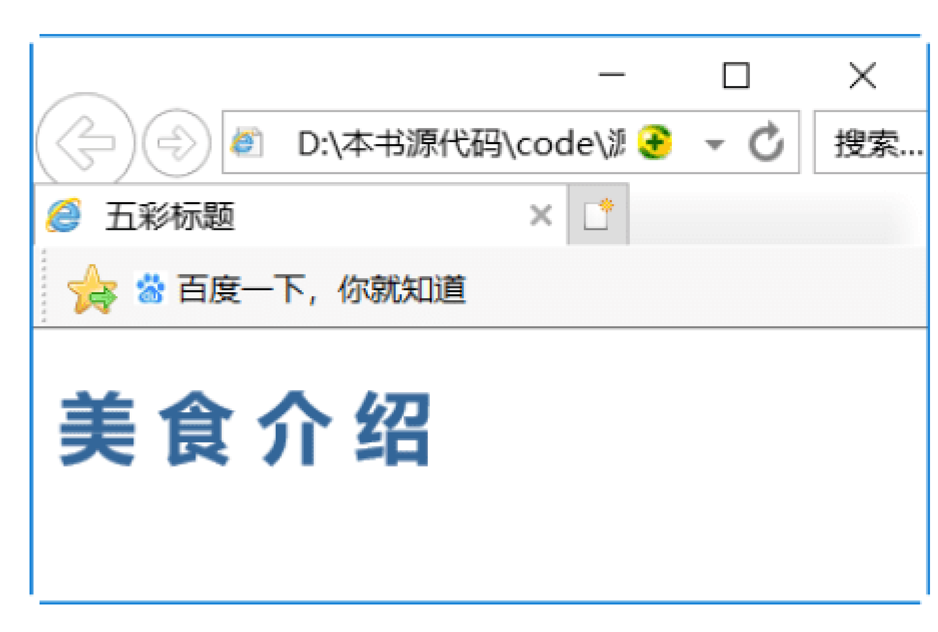


图 9-29　添加文本修饰标记

05 加入灰色边框。

为标题添加边框，代码如下：

```
padding-bottom:4px;              /*设置边框与文字的距离为4px*/
border-bottom:2px solid #ccc;    /*设置边框的颜色*/
```

在 IE 11.0 中浏览效果如图 9-30 所示，可以看到“美食介绍”文字下面添加一个边框，边框和文字距离是 4 像素。

06 增加背景图。

使用 CSS 样式为标记<h1>添加背景图片，代码如下：

```
background:url(01.jpg) repeat-x bottom;   /*设置背景图片和水平平铺模式*/
```

在 IE 11.0 中浏览效果如图 9-31 所示，可以看到“美食介绍”文字下面添加一个背景图片，图片在水平（X）轴方向进行平铺。

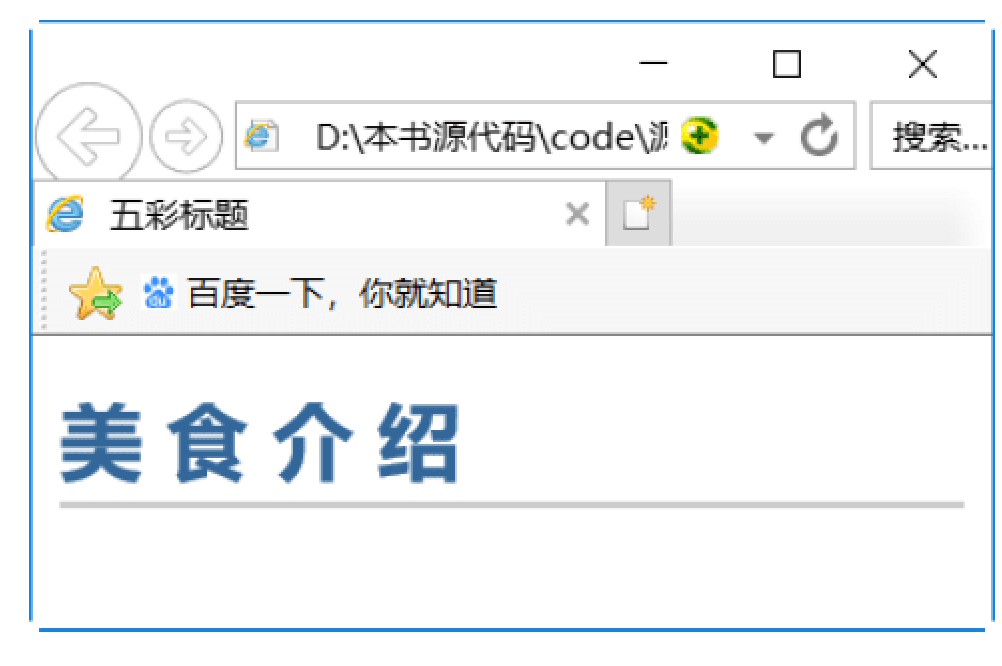


图 9-30　添加边框样式

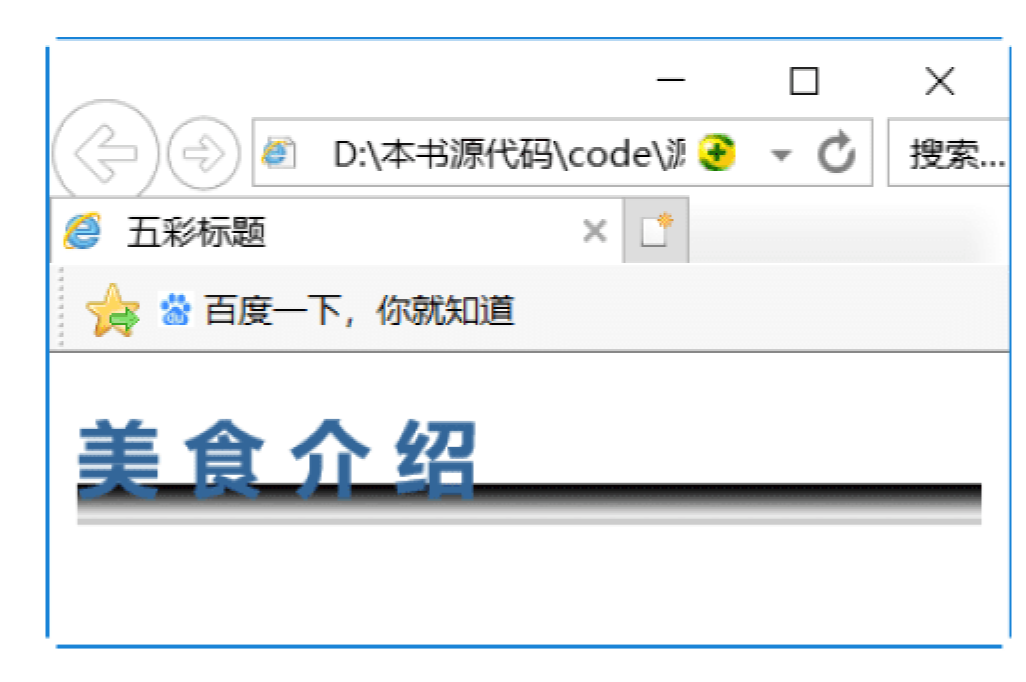


图 9-31　添加背景

07 定义标题宽度。

使用 CSS 属性将背景图变小，使其正好符合 4 个字体的宽度，代码如下：

```
width:120px;
```

在 IE 11.0 中浏览效果如图 9-32 所示，可以看到“美食介绍”文字下面背景图缩短，正好和字体宽度相同。

08 定义字体颜色。

在 CSS 样式中为每个字定义颜色，代码如下。

```
.c1{
   color:#B3EE3A;    /*设置第1个字的颜色*/
}
.c2{
   color:#71C671;    /*设置第2个字的颜色*/
}
.c3{
   color:#00F5FF;    /*设置第3个字的颜色*/
}
.c4{
   color:#00EE00;    /*设置第4个字的颜色*/
}
```

在 IE 11.0 中浏览效果如图 9-33 所示，可以看到每个字体显示不同的颜色，加上背景色共有 5 种颜色。

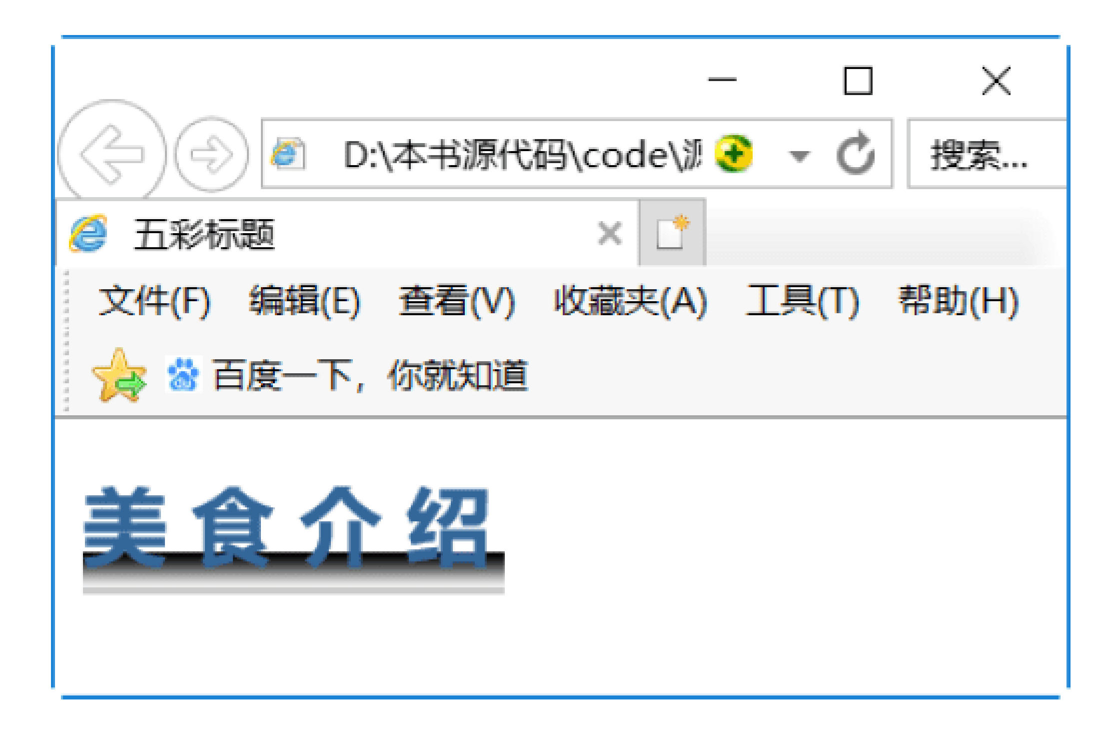


图 9-32 定义宽度

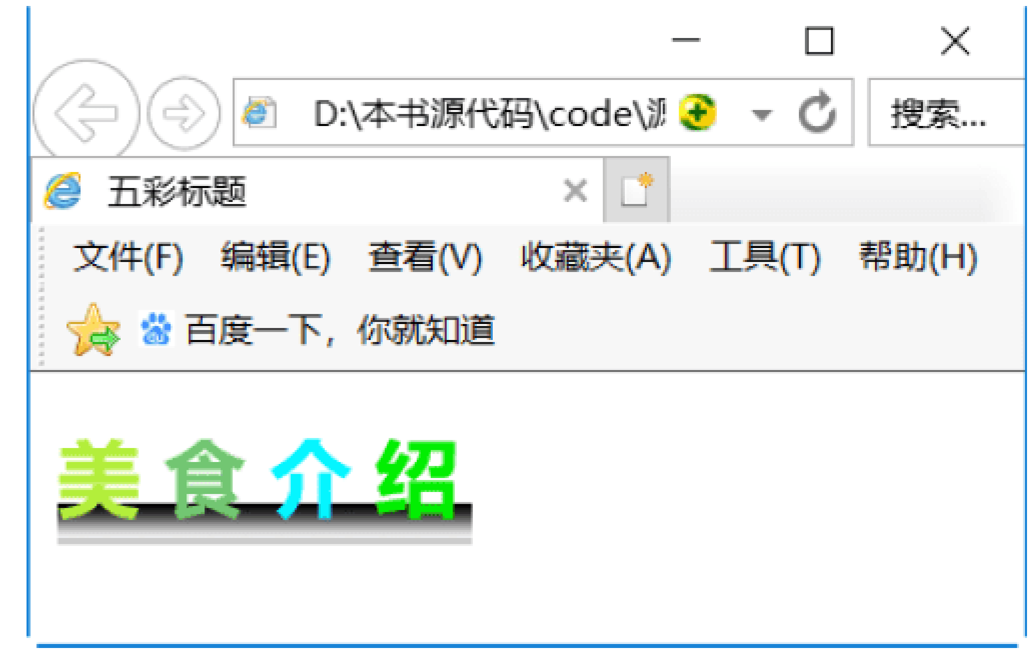


图 9-33 定义字体颜色

9.7 综合实例 2——制作新闻菜单

网上浏览新闻，是每个上网者都喜欢做的事情。一个布局合理，样式美观大方的新闻菜单是吸引人的主要途径之一。本实例使用 CSS 控制 HTML 标记创建新闻菜单，具体步骤如下：

01 分析需求。

创建一个新闻菜单需要包含两个部分：一个是父菜单，用来表明新闻类别；另一个是子菜单，介绍具体的新闻消息。菜单方式很多，可以用<table>创建，也可以用列表创建，同样也可以使用段落<p>创建。本实例采用<p>标记结合<div>创建。

02 分析局部和整体，构建 HTML 网页。

一个新闻菜单可以分为三个层次：新闻父菜单、新闻焦点和新闻子菜单，下面分别使用 div 创建。其 HTML 代码如下：

```
<!DOCTYPE html>
<html >
<head><title>导航菜单</title>
</head>
<body>
<div class="big">
<h2>时事热点 </h2>
<div class="up">
<a href="#">7月周周爬房团报名</a>
</div> <div class="down">
<p>·50万买下两居会员优惠 全世界大学排名 工薪阶层留学美国</p>
<p>·家电 | 买房上焦点打电话送礼 楼市松动百余项目打折</p>
```

```
<p>•财经 | 油价大跌  CPI新高 </p>
</div>
</div>
</body>
</html>
```

在 IE 11.0 中的显示效果如图 9-34 所示，可以看到一个标题、一个超链接和三个段落以普通样式显示，其布局只存在上下层次。



图 9-34　无 CSS 标记显示

03 添加 CSS 代码，修饰整体样式。

对于 HTML 页面，需要有一个整体样式。其代码如下：

```
<style>
*{/*全局选择器*/
padding:0px;
margin:0px;
}
body{
font-family:"宋体";          /*设置文本的字体样式*/
font-size:12px;             /*设置字体的大小*/
}
.big{
width:400px;                 /*设置边框的宽度*/
border:#33CCCC 1px solid; /*设置边框的颜色为浅绿色*/
}
</style>
```

在 IE 11.0 中的显示效果如图 9-35 所示，可以看到全局层会以边框显示，宽度为 400 像素，其颜色为浅绿色；文档内容中字体采用宋体，大小为 12 像素，并且定义内容和层之间空隙为 0，层和层之间空隙为 0。

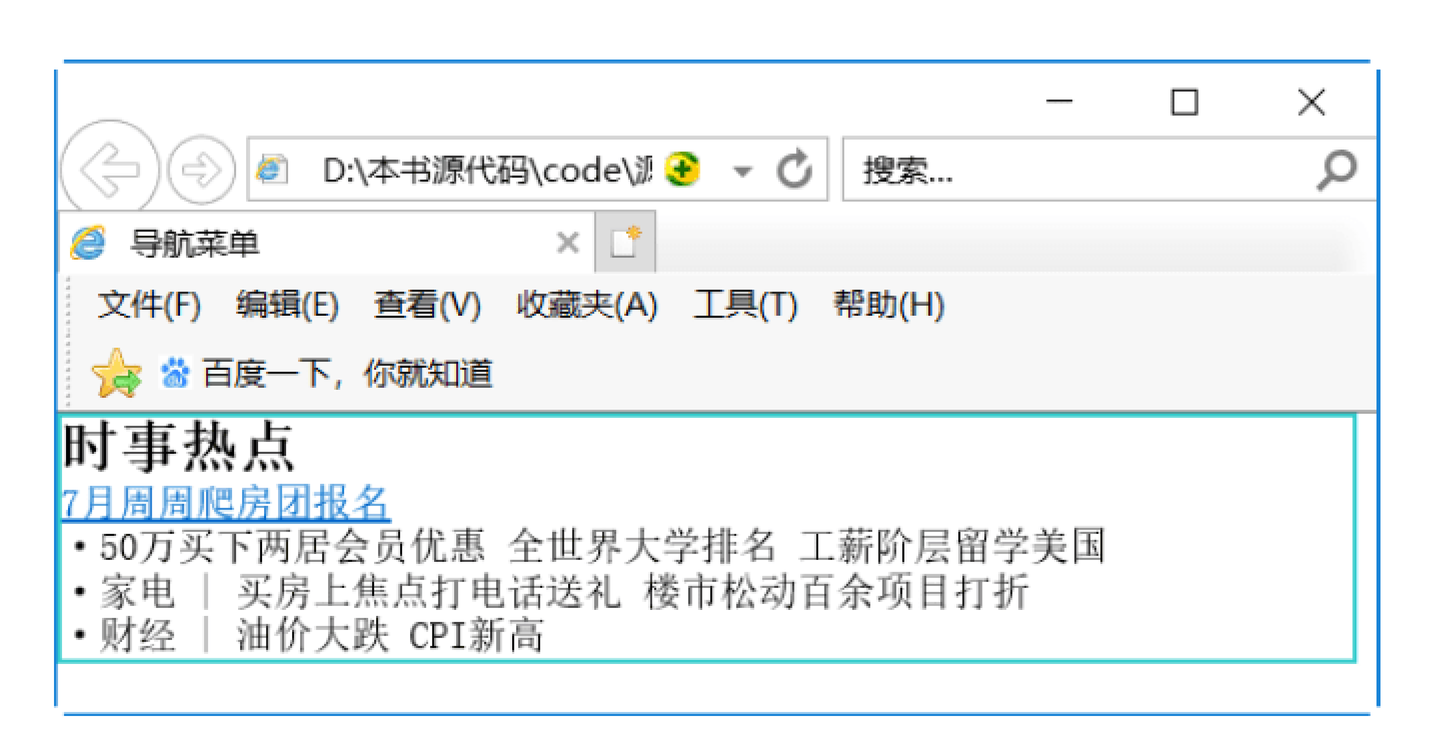


图 9-35　整体样式添加

04 添加 CSS 代码，修饰新闻父菜单。

对新闻父类菜单进行 CSS 控制。其代码如下：

```
h2{background-color:olive;      /*设置背景颜色*/
    display:block;               /*设置方框的显示方式*/
    width:400px;                /*设置方框的宽度*/
    height:18px;                /*设置方框的高度*/
    line-height:18px;          /*设置字体的行高*/
    font-size:14px;}           /*设置字体的大小*/
```

在 IE 11.0 中的显示效果如图 9-36 所示，可以看到标题“时事热点”会以矩形方框显示，其背景色为橄榄色，字体大小为 14 像素，行高为 18 像素。

05 添加 CSS 菜单，修饰子菜单。

```
.up{padding-bottom:5px;  /*设置下边距的大小*/
    text-align:center;   /*设置文本居中显示*/
}
p{line-height:20px;}    /*设置文本的行高*/
```

在 IE 11.0 中的显示效果如图 9-37 所示，可以看到超链接“7 月周周爬房团报名”居中显示，所有段落之间间隙增大。

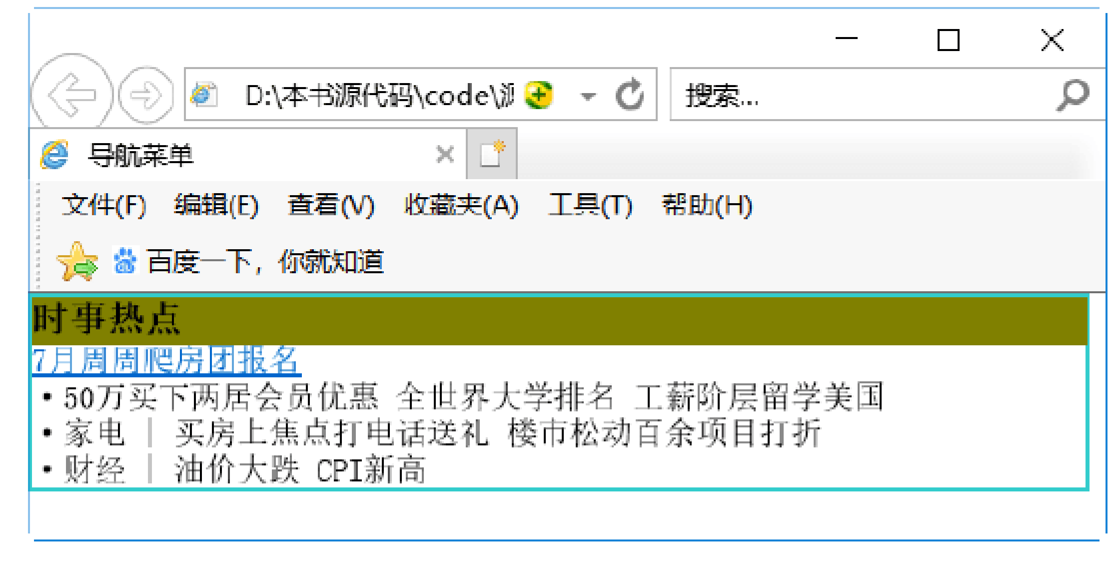


图 9-36　修饰超级链接图

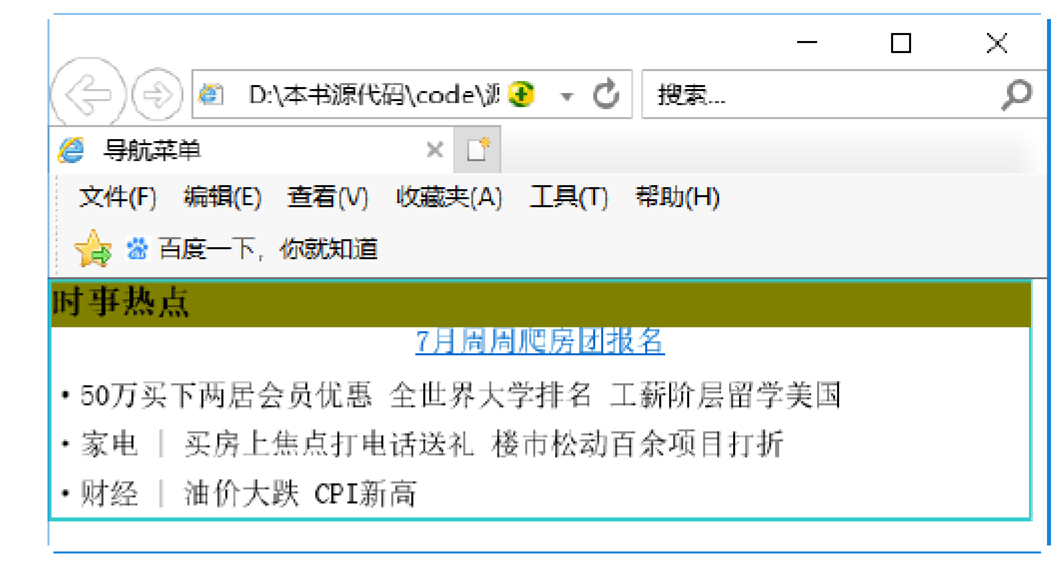


图 9-37　子菜单样式显示

06 添加 CSS 菜单，修饰超级链接。

```
a{ /*设置超级链接文字的样式*/
font-size:16px;
    font-weight:800;
    text-decoration:none;
    margin-top:5px;
    display:block;}
a:hover{/*设置鼠标放置超级链接文字上的样式*/
color:#FF0000;
    text-decoration:underline;}
```

在 IE 11.0 中的显示效果如图 9-38 所示。可以看到超链接“7 月周周爬房团报名”字体变大，加粗，并且无下画线显示，将鼠标指针放在此超级链接上，会以红色字体显示，并且下面带有下画线。

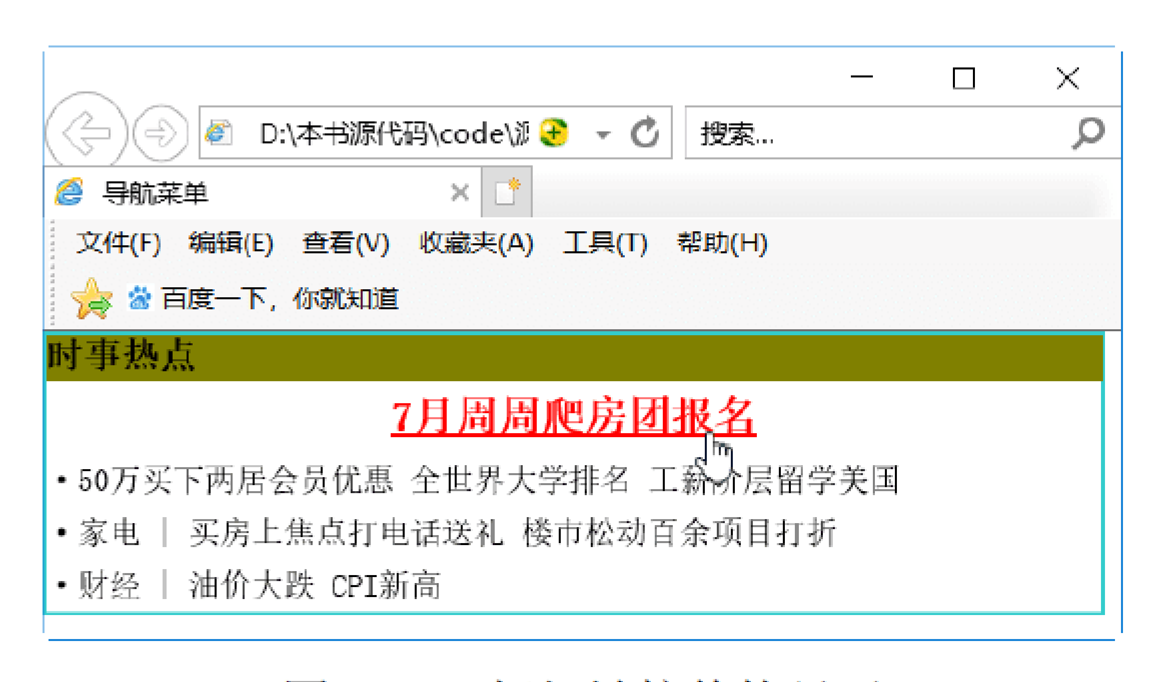


图 9-38 超级链接修饰显示

9.8 专家解惑

1. CSS 定义字体在不同浏览器大小不一样？

例如使用 font-size:14px 定义的宋体文字，在 IE 下实际高是 16 像素，下空白是 3 像素，Firefox 浏览器下实际高是 17 像素、上空 1 像素、下空 3 像素。其解决办法是在文字定义上设置 line-height，并确保所有文字都有默认的 line-height 值。

2. CSS 在网页制作中一般有 4 种方式的用法，那么具体在使用时该采用哪种用法？

当有多个网页需要用到的 CSS 时，就采用外链 CSS 文件的方式，这样网页的代码将大大减少，修改起来也非常方便；只在单个网页中使用的 CSS 时，就采用文档头部方式；只有在网页一、两个地方用到的 CSS 时，就采用行内插入方式。

3. CSS 的行内样式、内嵌样式和链接样式可以在一个网页中混用吗？

3 种用法可以混用且不会造成混乱，这也是它为什么称之为“层叠样式表”的原因。浏览器在显示网页时是这样处理的：先检查有没有行内插入式 CSS，有就执行了，针对本句的其他 CSS 就不去管它了；其次检查内嵌方式的 CSS，有就执行了；在前两者都没有的情况下再检查外链文件方式的 CSS。因此可以看出，3 种 CSS 的执行优先级是行内样式、内嵌样式、链接样式。

第10章　CSS3字体与段落属性

常见的网站、博客通常使用文字或图片来阐述自己的观点，其中文字是传递信息的主要手段。而美观大方的网站或博客，需要使用 CSS 样式修饰。设置文本样式是 CSS 技术的基本使命，通过 CSS 文本标记语言，可以设置文本的样式和粗细等。

10.1　字体属性

一个杂乱无序、堆砌而成的网页，会使人产生枯燥无味、望而止步的感觉，而一个美观大方的网页，会给人以美轮美奂、流连忘返的感觉。这美观大方的效果，都是使用 CSS 字体样式来设置的。通过对本节内容的学习，相信读者可以设计出令人流连忘返的网页。

10.1.1　字体 font-family

font-family 属性用于指定文字字体类型，如宋体、黑体、隶书、Times New Roman 等，即在网页中展示字体不同的形状。具体的语法格式如下：

```
{font-family:name}
{font-family:cursive|fantasy|monospace|serif|sans-serif}
```

从语法格式上可以看出，font-family 有两种声明方式。第一种声明方式使用 name 字体名称，按优先顺序排列，以逗号隔开，如果字体名称包含空格，则应使用引号括起；第二种声明方式使用所列出的字体序列名称，如果使用 fantasy 序列，将提供默认字体序列。在 CSS3 中，比较常用的是第一种声明方式。

【例 10.1】（实例文件：ch10\10.1.html）

```
<!DOCTYPE html>
<html>
<head>
<style type=text/css>
p{font-family:黑体}
</style>
</head>
```

```
<body>
<p align=center>天行健，君子以自强不息。</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-1 所示，可以看到文字居中并以黑体显示。

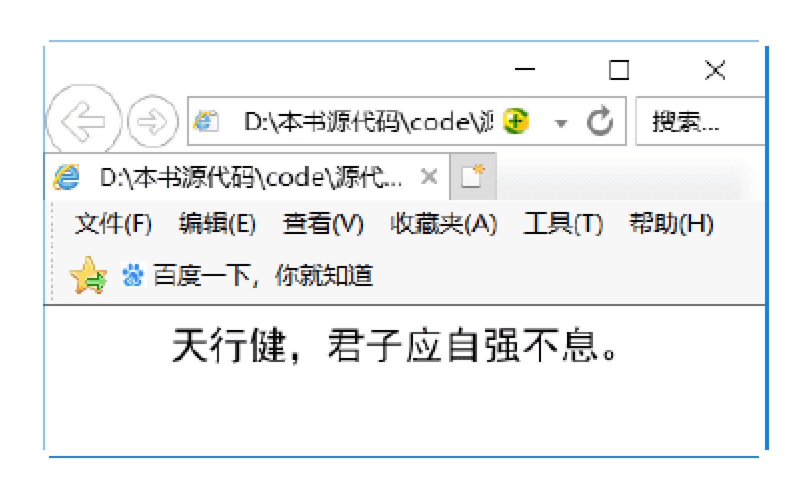


图 10-1 字体类型显示

在字体显示时，如果指定一种特殊字体类型，而在浏览器或操作系统中该类型不能正确获取，就可以通过 font-family 预设多找字体类型。font-family 属性可以预置多个供页面使用的字体类型，即字体类型序列，其中每种字体类型之间使用逗号隔开。如果前面的字体类型不能够正确显示，则系统将自动选择后一种字体类型，依次类推。

所以，在设计页面时一定要考虑字体的显示问题。为了保证页面达到预计的效果，最好提供多种字体类型，而且最好以最基本的字体类型作为最后一个，其样式设置如下：

```
P{
font-family:华文彩云,黑体,宋体
}
```

当 font-family 属性值中的字体类型由多个字符串和空格组成时（如 Times New Roman），该值就需要使用双引号引起来。

```
p{
   font-family:"Times New Roman"
}
```

10.1.2 字号 font-size

一个网页中，标题通常使用较大字体显示，用于吸引人注意，小字体用来显示正常内容，大小字体结合形成的网页，既吸引了人的眼球，又提高了人的阅读效率。

在 CSS3 新规定中，通常使用 font-size 设置文字大小。其语法格式如下：

```
   {font-size:数值
|inherit|xx-small|x-small|small|medium|large|x-large|xx-large|larger|
smaller|length}
```

其中，通过数值来定义字体大小，例如用 font-size:10px 的方式定义字体大小为 10 像素。此外，还可以通过其他属性值定义字体的大小，各属性值含义如表 10-1 所示。

表 10-1　字号属性值

属性值	说明
xx-small	绝对字体尺寸。根据对象字体进行调整。最小
x-small	绝对字体尺寸。根据对象字体进行调整。较小
small	绝对字体尺寸。根据对象字体进行调整。小
medium	默认值。绝对字体尺寸。根据对象字体进行调整。正常
large	绝对字体尺寸。根据对象字体进行调整。大
x-large	绝对字体尺寸。根据对象字体进行调整。较大
xx-large	绝对字体尺寸。根据对象字体进行调整。最大
larger	相对字体尺寸。相对于父对象中字体尺寸进行相对增大。使用成比例的 em 单位计算
smaller	相对字体尺寸。相对于父对象中字体尺寸进行相对减小。使用成比例的 em 单位计算
length	百分数或由浮点数字和单位标识符组成的长度值，不可为负值。其百分比取值是基于父对象中字体的尺寸

【例 10.2】（实例文件：ch10\10.2.html）

```
<!DOCTYPE html>
<html>
<body>
<div style="font-size:10pt">上级标记大小
<p style="font-size:small">小</p>
<p style="font-size:larger">大</p>
<p style="font-size:x-small">小</p>
<p style="font-size:x-larger">大</p>
<p style="font-size:50%">子标记</p>
<p style="font-size:25pt">子标记</p>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-2 所示，可以看到网页中的文字被设置成不同的大小，其设置方式采用了绝对数值、关键字和百分比等形式。

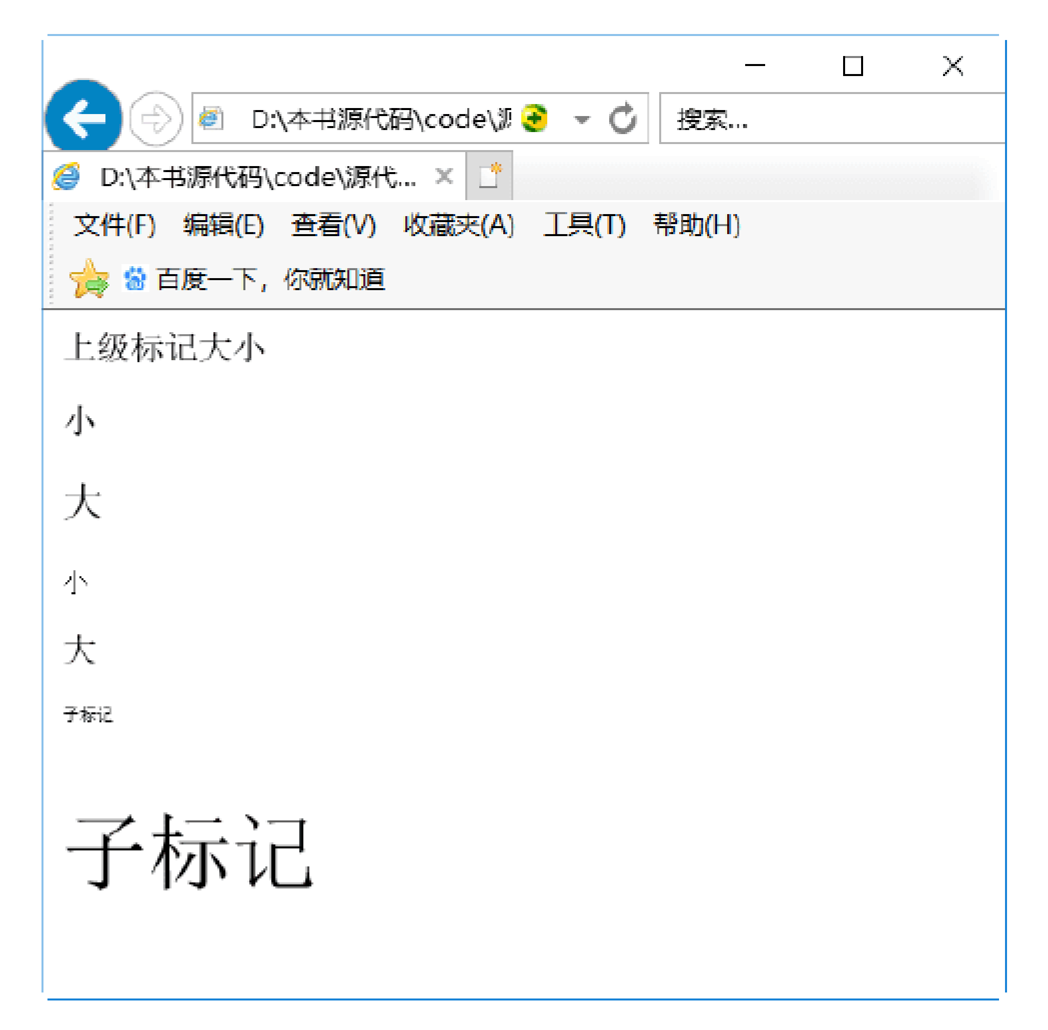


图 10-2　字体大小显示

在上面例子中，font-size 字体大小为 50%时，其比较对象是上一级标签中的 10pt。同样还可以使用 inherit 值，直接继承上级标记的字体大小，代码如下：

```
<div style="font-size:50pt">上级标记
<p style="font-size: inherit ">继承</p>
</div>
```

10.1.3　字体风格 font-style

font-style 通常用来定义字体风格，即字体的显示样式。在 CSS3 新规定中，语法格式如下：

```
font-style:normal|italic|oblique|inherit
```

其属性值有 4 个，具体含义如表 10-2 所示。

表 10-2　字体风格属性值

属性值	含义
normal	默认值。浏览器显示一个标准的字体样式
italic	浏览器会显示一个斜体的字体样式
oblique	浏览器会显示一个倾斜的字体样式
inherit	规定应该从父元素继承字体样式

【例 10.3】（实例文件：ch10\10.3.html）

```
<!DOCTYPE html>
<html>
<body>
```

```
<p style="font-style:italic">梅花香自苦寒来</p>
<p style="font-style:normal">梅花香自苦寒来</p>
<p style="font-style:oblique">梅花香自苦寒来</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-3 所示，可以看到文字分别显示不同的样式。

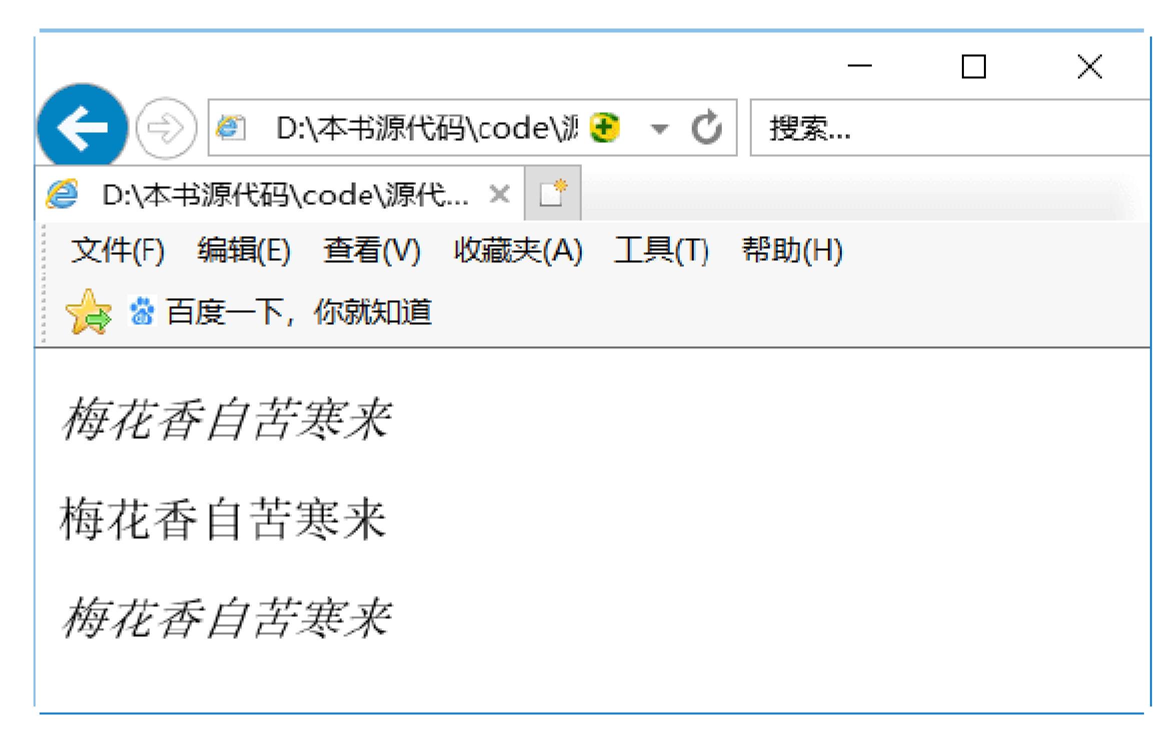


图 10-3　字体风格显示

10.1.4　加粗字体 font-weight

通过设置字体粗细，可以让文字显示不同的外观。通过 CSS3 中的 font-weight 属性可以定义字体的粗细程度，其语法格式如下：

```
{font-weight:100-900|bold|bolder|lighter|normal;}
```

font-weight 属性有 13 个属性值，分别是 bold、bolder、lighter、normal、100~900。如果没有设置该属性，则使用其默认值 normal。属性值设置为 100~900，值越大，加粗的程度就越高，其具体含义如表 10-3 所示。

表 10-3　加粗字体属性值

属性	描述
bold	定义粗体字体
bolder	定义更粗的字体，相对值
lighter	定义更细的字体，相对值
normal	默认，标准字体

浏览器默认的字体粗细是 400，也可以通过参数 lighter 和 bolder 使字体在原有基础上显得更细或更粗。

【例 10.4】（实例文件：ch10\10.4.html）

```
<!DOCTYPE html>
<html>
<body>
<p style="font-weight:bold">学习雷锋好榜样(bold)</p>
<p style="font-weight:bolder">学习雷锋好榜样(bolder)</p>
<p style="font-weight:lighter">学习雷锋好榜样(lighter)</p>
<p style="font-weight:normal">学习雷锋好榜样(normal)</p>
<p style="font-weight:100">学习雷锋好榜样(100)</p>
<p style="font-weight:400">学习雷锋好榜样(400)</p>
<p style="font-weight:900">学习雷锋好榜样(900)</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-4 所示，可以看到文字以不同方式加粗，其中使用了关键字加粗和数值加粗。

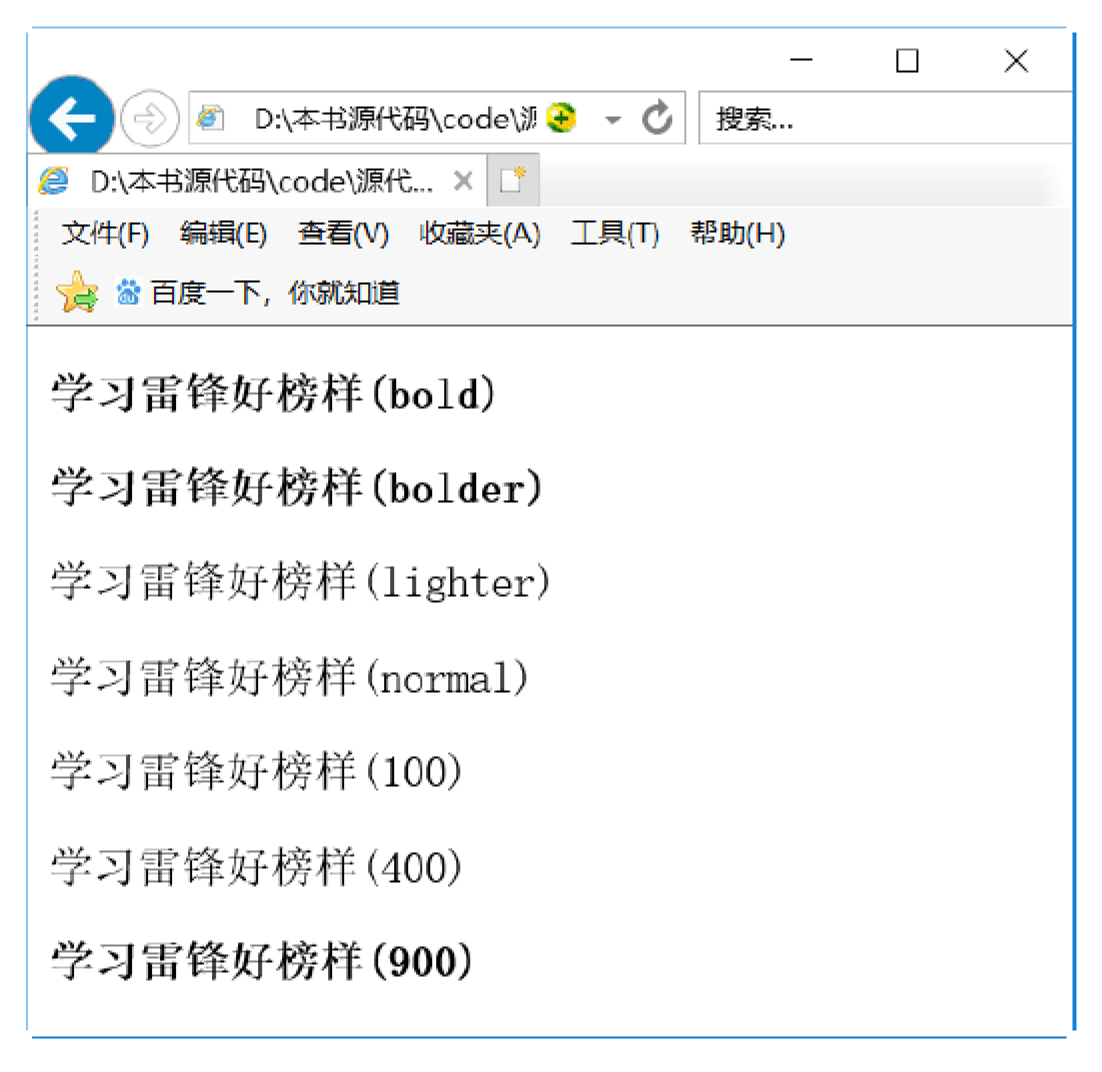


图 10-4　字体粗细显示

10.1.5　小写字母转为大写字母 font-variant

font-variant 属性设置大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写。在 CSS3 中，其语法格式如下：

```
{font-variant:normal|small-caps|inherit}
```

font-variant 有 3 个属性值，分别是 normal、small-caps 和 inherit，其具体含义如表 10-4 所示。

表 10-4　各属性值含义

属性值	说明
normal	默认值，浏览器会显示一个标准的字体
small-caps	浏览器会显示小型大写字母的字体
inherit	规定应该从父元素继承 font-variant 属性的值

【例 10.5】（实例文件：ch10\10.5.html）

```
<!DOCTYPE html>
<html>
<body>
<p style="font-variant:normal">Happy BirthDay to You</p>
<p style="font-variant:small-caps">Happy BirthDay to You</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-5 所示，可以看到字母以大写形式显示。

图中通过对两个属性值产生的效果进行比较可以看到，设置为 normal 属性值的文本以正常文本显示，而设置为 small-caps 属性值的文本中有稍大的大写字母，也有小的大写字母。也就是说，使用了 small-caps 属性值的段落文本全部变成了大写，只是大写字母的尺寸不同。

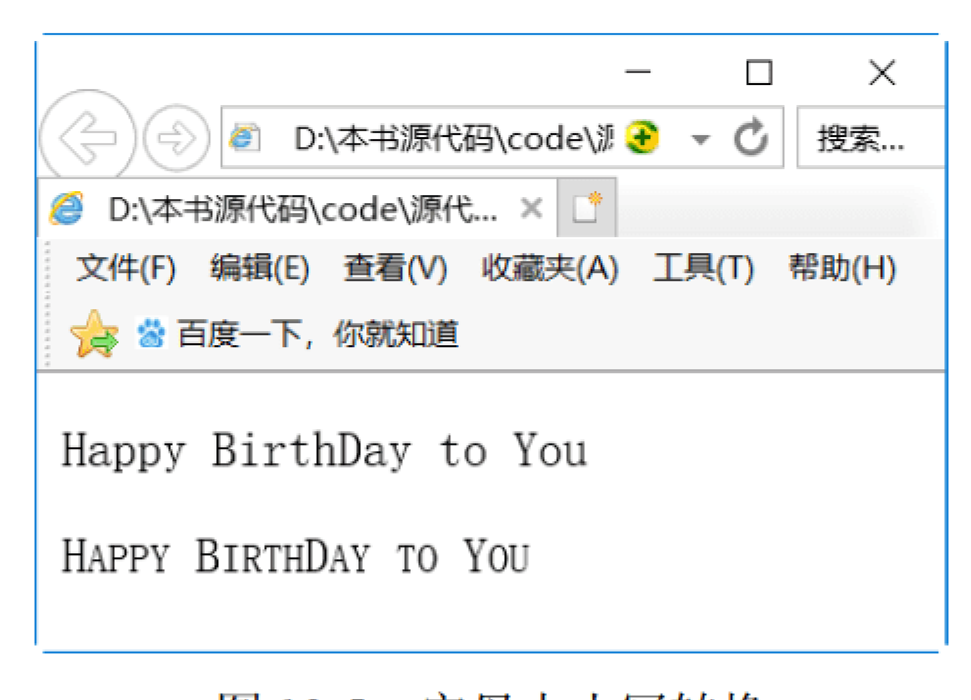


图 10-5　字母大小写转换

10.1.6　字体复合属性 font

在设计网页时，为了使网页布局合理且文本规范，对字体设计需要使用多种属性，如定义字体粗细、定义字体大小等。但是多个属性分别书写相对比较麻烦，CSS3 样式表提供的 font 属性就解决了这一问题。

font 属性可以一次性地使用多个属性的属性值定义文本字体，其语法格式如下：

```
{font:font-style font-variant font-weight font-size font-family}
```

font 属性中的属性排列顺序是 font-style、font-variant、font-weight、font-size 和 font-family，各属性的属性值之间使用空格隔开，但是如果 font-family 属性要定义多个属性值，则需使用逗号“，”隔开。

属性排列中，font-style、font-variant 和 font-weight 这 3 个属性值是可以自由调换的。而 font-size 和 font-family 则必须按照固定的顺序出现，如果这两个的顺序不对或缺少一个，那么整条样式规则可能因此会被忽略。

【例 10.6】（实例文件：ch10\10.6.html）

```
<!DOCTYPE html>
<html>
<head>
<style type=text/css>
p{
   font:normal small-caps bolder 20pt "Cambria","Times New Roman",宋体
}
</style>
</head>
<body>
<p>读书和学习是在别人思想和知识的帮助下，建立起自己的思想和知识。</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-6 所示，可以看到文字被设置成宋体并加粗。

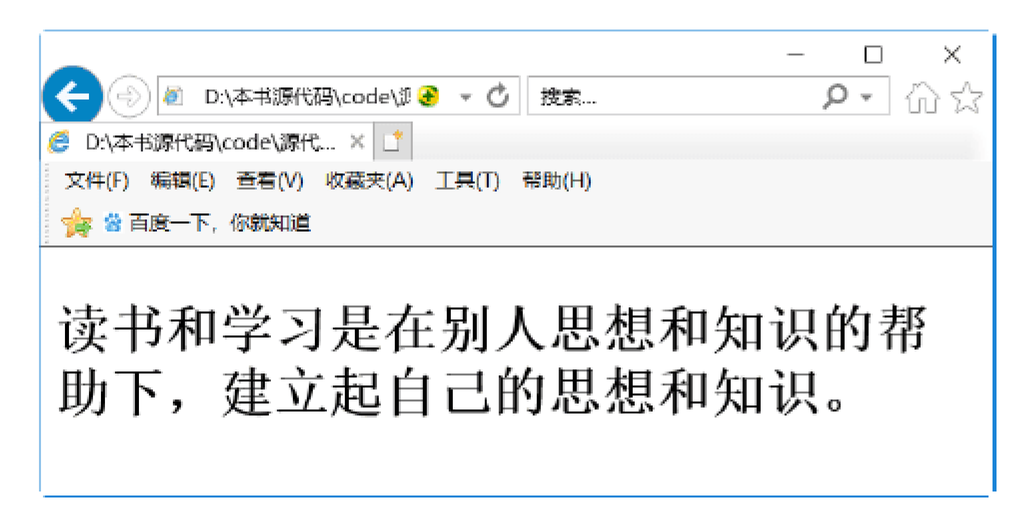


图 10-6 复合属性 font 显示

10.1.7 字体颜色 color

没有色彩的网页是枯燥而没有生机的，这就意味着一个优秀的网页设计者不仅要能够合理安排页面布局，而且还要具有一定的色彩视觉和色彩搭配能力，这样才能够使网页更加精美，具有表现力，给浏览者以亲切感。

在 CSS3 样式中，通常使用 color 属性来定义颜色。其属性值通常使用的设置域如表 10-5 所示。

表 10-5　属性值设置域

属性值	说明
color_name	规定属性值为颜色名称的颜色（如 red）
hex_number	规定属性值为十六进制值的颜色（如#ff0000）
rgb_number	规定属性值为 rgb 代码的颜色（如 rgb(255,0,0)）
inherit	规定应该从父元素继承颜色
hsl_number	规定属性值为 HSL 代码的颜色（如 hsl(0,75%,50%)），此为 CSS3 新增加的颜色表现方式
hsla_number	规定属性值为 HSLA 代码的颜色（如 hsla(120,50%,50%,1)），此为 CSS3 新增加的颜色表现方式
rgba_number	规定属性值为 RGBA 代码的颜色（如 rgba(125,10,45,0.5)），此为 CSS3 新增加的颜色表现方式

【例 10.7】（实例文件：ch10\10.7.html）

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
body{color:red}                    /*设置body标记的颜色*/
h1{color:#00ff00}                  /*设置h1标记的颜色*/
p.ex{color:rgb(0,0,255)}           /*设置组合选择器p.ex的颜色*/
p.hs{color:hsl(0,75%,50%)}         /*设置组合选择器p.hs的颜色*/
p.ha{color:hsla(120,50%,50%,1)}    /*设置组合选择器p.ha的颜色*/
p.ra{color:rgba(125,10,45,0.5)}    /*设置组合选择器p.ra的颜色*/
</style>
</head>
<body>
<h1>这是标题1</h1>
<p>这是一段普通的段落。请注意，该段落的文本是红色的。在 body 选择器中定义了本页面中的默认文本颜色。</p>
<p class="ex">该段落定义了 class="ex"。该段落中的文本是蓝色的。</p>
<p class="hs">此处使用了CSS3中的新增加的HSL函数，构建颜色。</p>
<p class="ha">此处使用了CSS3中的新增加的HSLA函数，构建颜色。</p>
<p class="ra">此处使用了CSS3中的新增加的RGBA函数，构建颜色。</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-7 所示，可以看到文字以不同颜色显示，并采用了不同的颜色取值方式。

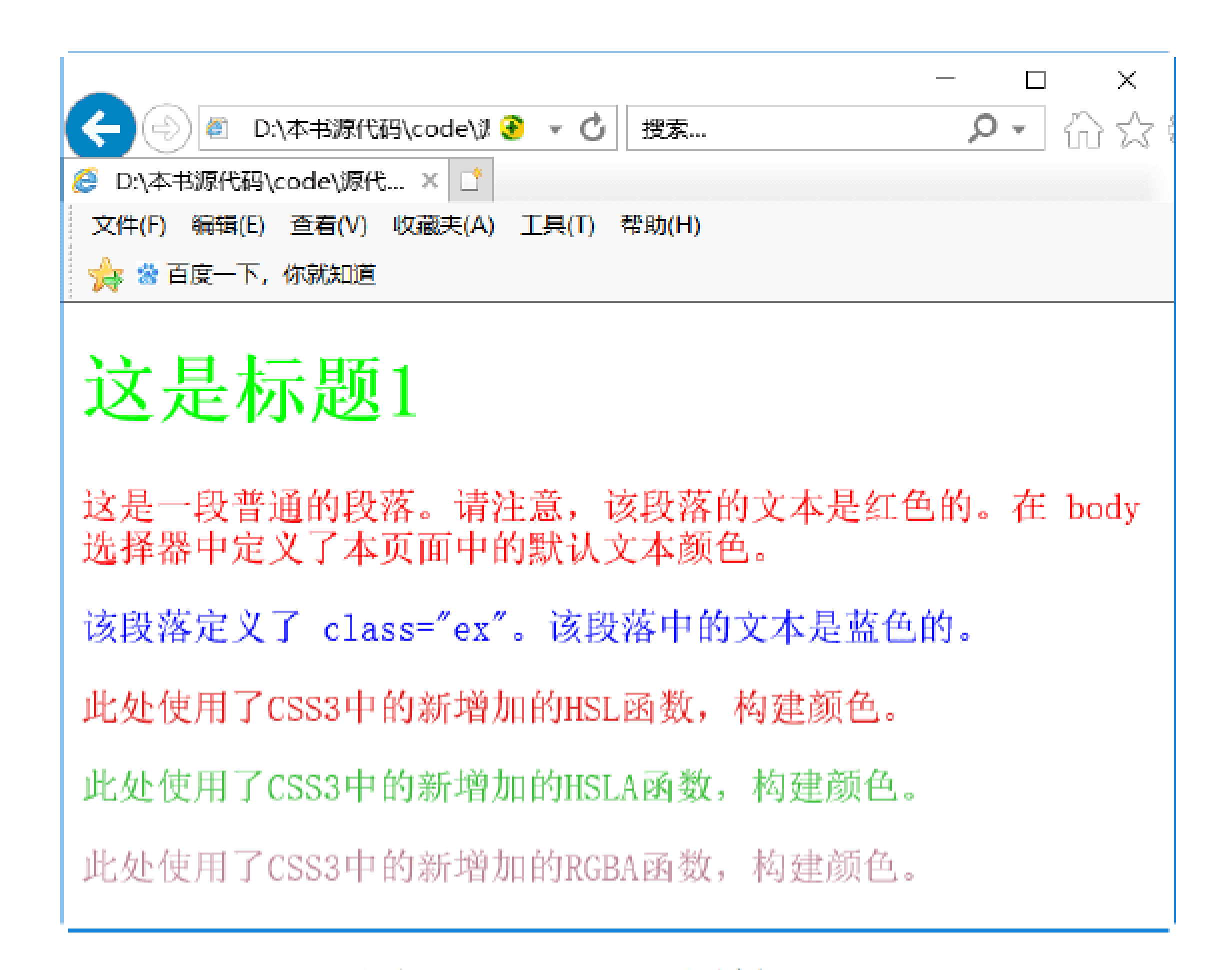


图 10-7　color 属性显示



在本实例中，使用了 3 个 CSS3 中新增加的表现形式，分别是 HSL（）、HSLA 和 RGBA（），这 3 个函数在 Firefox 浏览器中支持，但 IE 浏览器尚不支持。

10.2　文本高级样式

对于一些有特殊要求的文本（如文字存在阴影），字体种类会发生变化。如果再使用上面所介绍的 CSS 样式进行定义，其结果就不会得到正确显示，这时就需要一些特定的 CSS 标记来完成这些要求。

10.2.1　阴影文本 text-shadow

在显示字体时，有时根据要求需要给文字添加阴影效果并为文字阴影添加颜色，以增强网页整体表现力，这时就需要用到 CSS3 样式中的 text-shadow 属性。实际上在 CSS 2 中，W3C 就已经定义了 text-shadow 属性，只是 CSS3 又重新定义了它，并为其增加了不透明度效果。

text-shadow 属性有 4 个属性值，最后两个是可选的，第一个属性值表示阴影的水平位移，可取正负值；第二个属性值表示阴影垂直位移，可取正负值；第三个属性值表示阴影模糊半径，不可为负值，该值可选；第 4 个个属性值表示阴影颜色值，该值可选。语法格式如下：

```
text-shadow: length length opacity color
```

【例 10.8】（实例文件：ch10\10.8.html）

```
<!DOCTYPE html>
<html>
<body>
<p align=center style="text-shadow:0.1em 2px 6px blue;font-size:80px;">
```

```
这是TextShadow的阴影效果</p>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 10-8 所示，可以看到文字居中并带有阴影显示。

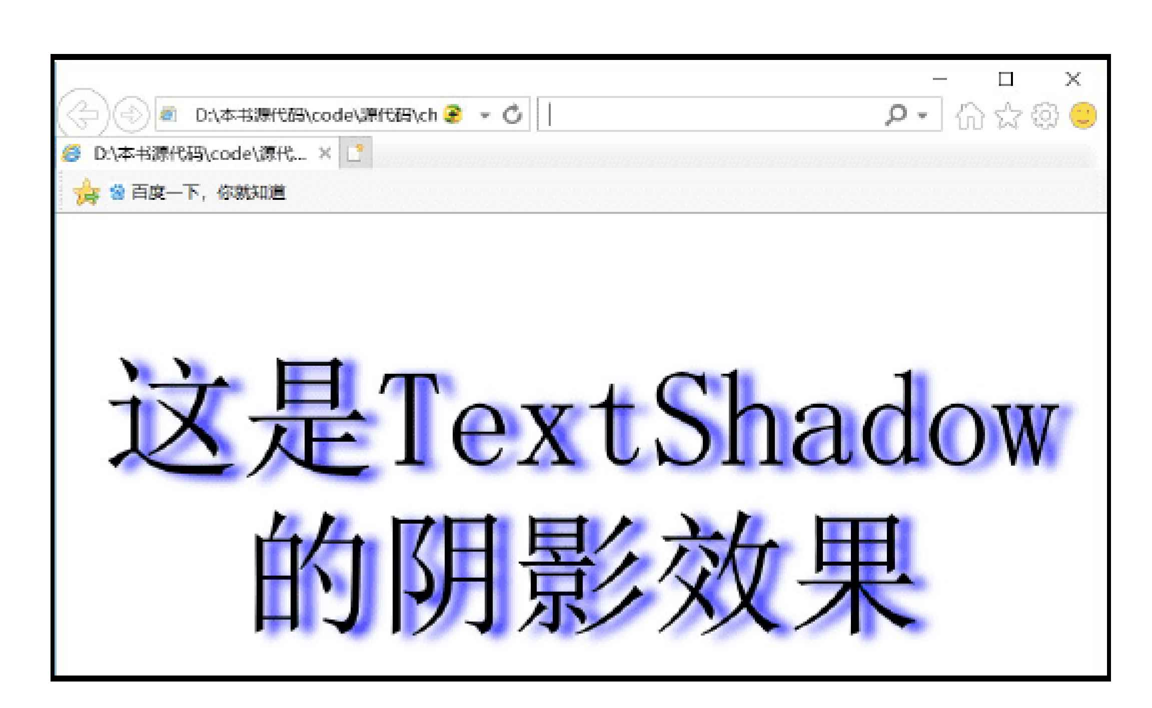


图 10-8　阴影显示结果图

通过上面的实例，可以看出阴影偏移由两个 length 值调整文本阴影的位移。第一个 length 值指定到文本右边的水平距离，负值会把阴影放置在文本左边；第二个 length 值指定到文本下边的垂直距离，负值会把阴影放置在文本上方。在阴影偏移之后，可以指定一个模糊半径。



模糊半径是一个长度值，它支持了模糊效果的范围，但如何计算效果的具体算法并没有指定。在阴影效果的长度值之前或之后，还可以指定一个颜色值。颜色值会被用作阴影效果的基础，如果没有指定颜色，那么将使用文本颜色来替代。

10.2.2　溢出文本 text-overflow

在网页显示信息时，如果指定显示区域宽度，而显示信息过长，其结果就是信息会撑破指定的信息区域，进而破坏整个网页布局。如果设置的信息显示区域过长，就会影响整体网页显示。以前，我们遇到这样的情况，通常使用 JavaScript 将超出的信息进行省略。现在，只需要使用 CSS3 新增的 text-overflow 属性，就可以解决这个问题。

text-overflow 属性用来定义当文本溢出时是否显示省略标记，即定义省略文本的显示方式，并不具备其他的样式属性定义。要实现溢出时产生省略号的效果还须定义强制文本在一行内显示（white-space:nowrap）及溢出内容为隐藏（overflow:hidden），只有这样才能实现溢出文本显示省略号的效果。

text-overflow 语法如下：

```
text-overflow: clip|ellipsis
```

其属性值含义如表 10-6 所示。

表 10-6　溢出文本属性值

属性值	说明
clip	不显示省略标记（...），而是简单的裁切条
ellipsis	当对象内文本溢出时显示省略标记（...）



这里需要特别说明的是，text-overflow 属性非常特殊，当设置的属性值不同时，其浏览器对 text-overflow 属性支持也不相同。当 text-overflow 属性值是 clip 时，现在主流的浏览器都支持，如果 text-overflow 属性是 ellipsis 时，除了 Firefox 5.0 浏览器暂不支持，其他主流浏览器都支持。

【例 10.9】（实例文件：ch10\10.9.html）

```
    <!DOCTYPE html>
    <html>
    <body>
    <style type="text/css">
    .test demo clip{text-overflow:clip; overflow:hidden; white-space:nowrap;
width:200px; background:#ccc;}
    .test demo ellipsis{text-overflow:ellipsis;overflow:hidden;white-space:
nowrap;width:200px;background:#ccc;}
    </style>
    <h2>text-overflow:clip</h2>
    <div class="test demo clip">
      不显示省略标记，而是简单的裁切条
    </div>
    <h2>text-overflow:ellipsis</h2>
    <div class="test demo ellipsis">
      显示省略标记，不是简单的裁切条
    </div>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 10-9 所示，可以看到文字在指定位置被裁切，ellipsis 属性以省略号形式出现。

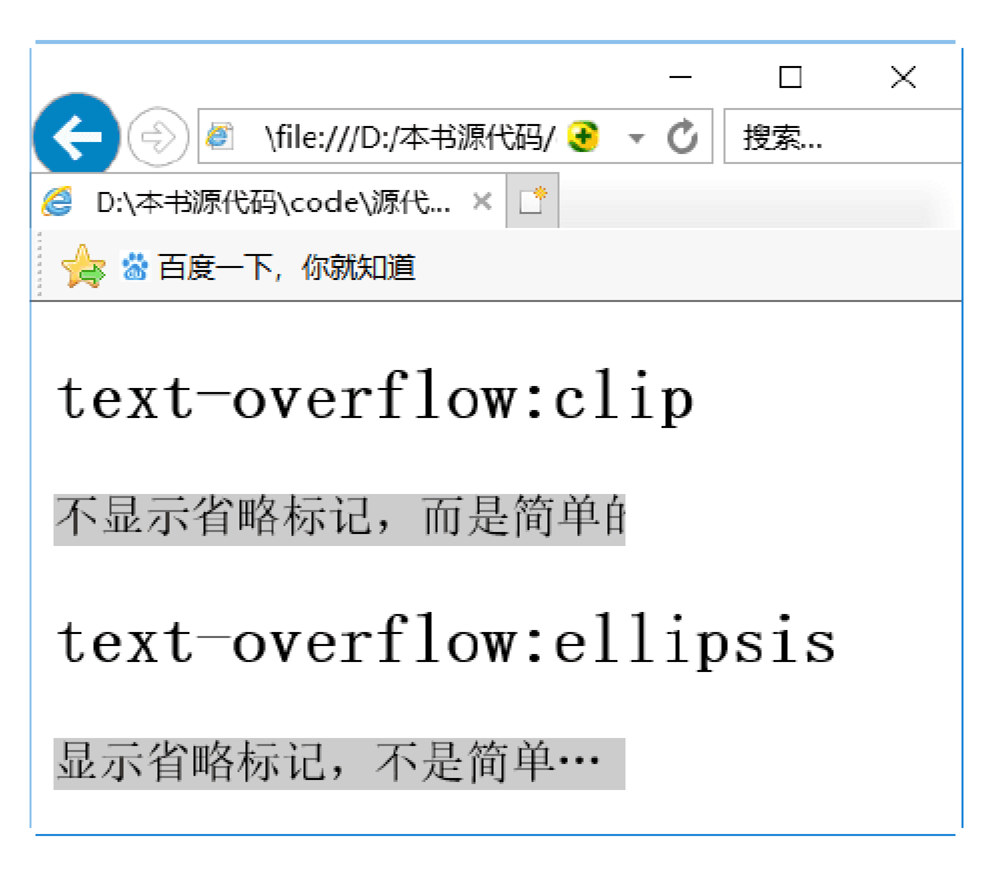


图 10-9　文本省略处理

10.2.3 控制换行 word-wrap

当在一个指定区域显示一整行文字时，如果文字在一行显示不完时，则需要进行换行；如果不进行换行，则会超出指定区域范围。此时我们可以采用 CSS3 中新增加的 word-wrap 文本样式来控制文本换行。

word-wrap 语法格式如下：

```
word-wrap: normal|break-word
```

其属性值含义比较简单，如表 10-7 所示。

表 10-7 控制换行属性值

属性值	说明
normal	允许内容顶开指定的边界
break-word	内容将在边界内换行。如果需要，词内换行（word-break）也会发生

【例 10.10】（实例文件：ch10\10.10.html）

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div{width:300px;word-wrap:break-word;border:1px solid #999999;}
</style>
</head>
<body>
<div>wordwrapbreakwordwordwrapbreakwordwordwrapbreakwordwordwrapbreakwo
rd</div>
<br />
<div>全中文的情况，全中文的情况，全中文的情况全中文的情况全中文的情况</div><br />
<div>This is all English,This is all English,This is all English,This is
all English</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-10 所示，可以看到文字在指定位置被控制换行。

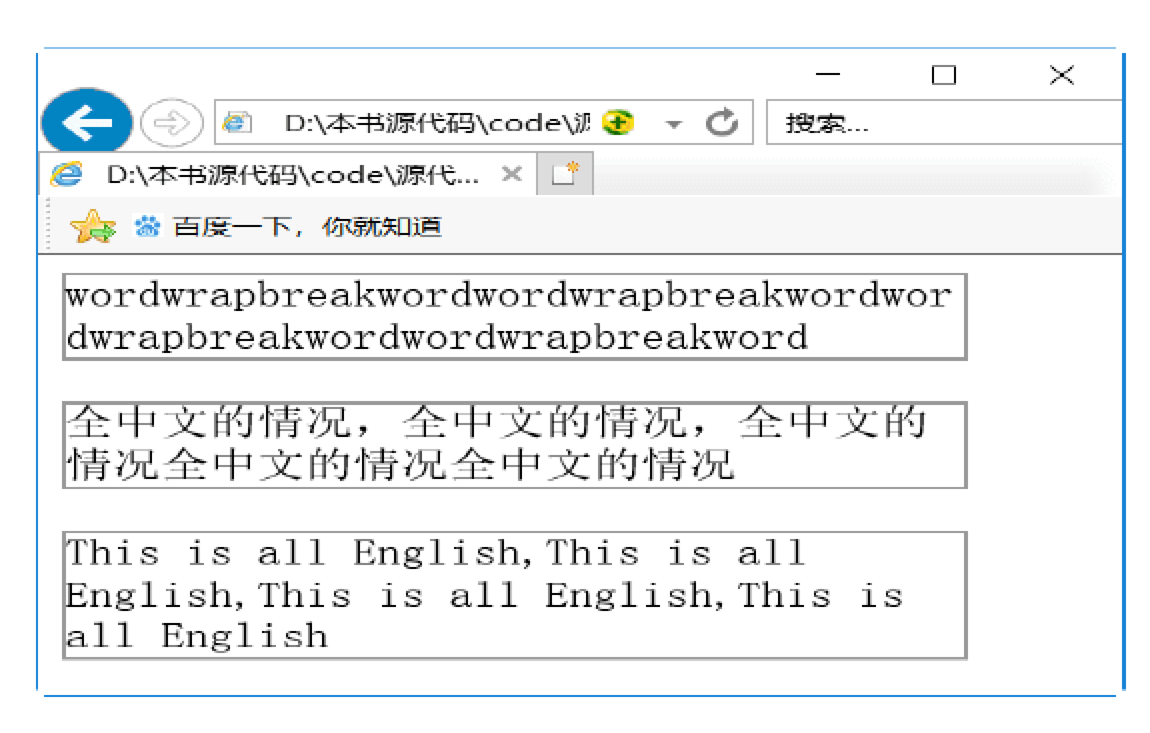


图 10-10 文本强制换行

可以看出，word-wrap 属性可以控制换行，当属性取值 break-word 时将强制换行。中文文本没有任何问题，英文语句也没有任何问题，但是对于长的英文字符串就不起作用，也就是说 break-word 属性只控制是否断词，而不是断字符。

10.2.4　保持字体尺寸不变 font-size-adjust

有时候在同一行的文字，由于所采用字体类型或修饰样式不同，而导致其字体尺寸不一样，整行文字看起来就显得很杂乱，此时需要 CSS3 属性 font-size-adjust 来处理。

font-size-adjust 用来定义整个字体序列中所有字体的大小是否保持同一个尺寸。其语法格式如下：

```
font-size-adjust:none|number
```

其属性值含义如表 10-8 所示。

表 10-8　保持字体尺寸不变属性值

属性值	说明
none	默认值，允许字体序列中每一字体遵守它自己的尺寸
number	为字体序列中所有字体强迫指定同一尺寸

【例 10.11】（实例文件：ch10\10.11 html）

```
<!DOCTYPE html>
<html>
<head>
<style>
.big{font-family:sans-serif;font-size:40pt;}
.a{font-family:sans-serif;font-size:15pt;font-size-adjust:1;}
.b{font-family:sans-serif;font-size:30pt;font-size-adjust:0.5;}
</style>
</head>
<body>
<p class="big"><span class="b">闻鸡起舞</span></p>
<p class="big"><span class="a">闻鸡起舞</span></p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-11 所示，可以看到同一行的字体大小相同。



图 10-11　尺寸一致显示

10.3　段落属性

网页由文字组成，而用来表达同一个意思的多个文字组合则称为段落，段落是文章的基本单位，同样也是网页的基本单位。段落的放置与效果的显示会直接影响到页面的布局及风格。CSS 样式表提供了文本属性来实现对页面中段落文本的控制。

10.3.1　单词间隔 word-spacing

单词之间的间隔如果设置合理，一是会给整个网页布局节省空间，二是可以给人赏心悦目的感觉，提高用户的阅读效率。在 CSS3 中，可以使用 word-spacing 直接定义指定区域或段落中字符之间的间隔。

word-spacing 属性用于设置词与词之间的间距。其语法格式如下：

```
word-spacing:normal|length
```

其中属性值 normal 和 length 含义如表 10-9 所示。

表 10-9　单词间隔属性值

属性值	说明
normal	默认，定义单词之间的标准间隔
length	定义单词之间的固定间隔，可以接受正值或负值

【例 10.12】（实例文件：ch10\10.12.html）

```
<!DOCTYPE html>
<html>
<body>
<p style="word-spacing:normal">Welcome to my home</p>
<p style="word-spacing:15px">Welcome to my home</p>
<p style="word-spacing:15px">欢迎来中国旅游</p>
</body>
```

```
</html>
```

在 IE 11.0 中浏览效果如图 10-12 所示，可以看到段落中单词以不同间隔显示。

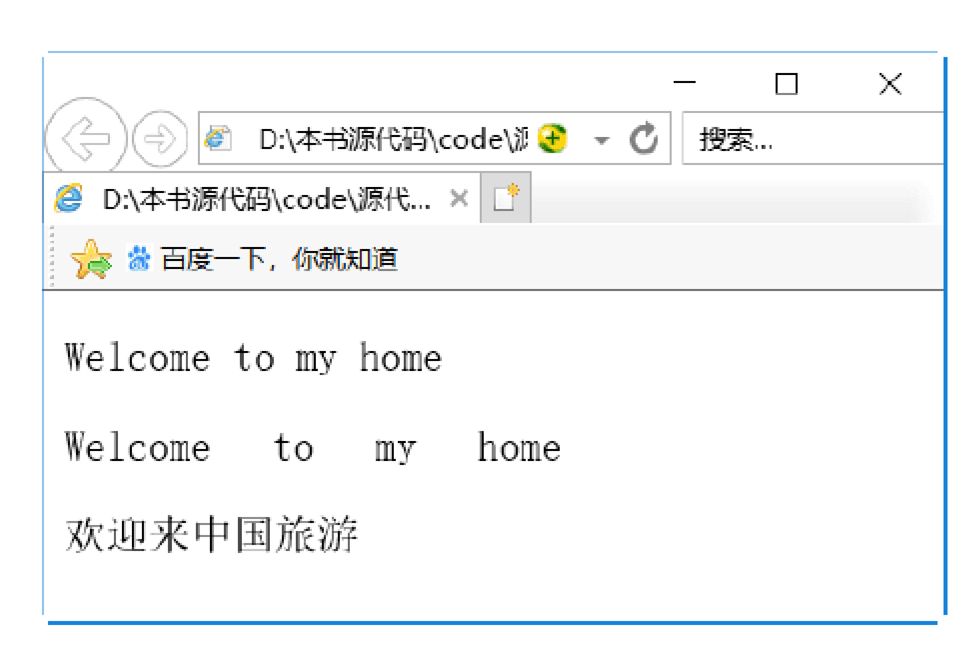


图 10-12 设置词间隔显示

从上面显示结果可以看出，word-spacing 属性不能用于设置文字之间的间隔。

10.3.2 字符间隔 letter-spacing

在一个网页中，还可能涉及多个字符文本。将字符文本之间的间距设置和词间隔保持一致，进而保持页面的整体性是网页设计者必须完成的。词与词之间可以通过 word-spacing 进行设置，那么字符之间使用什么设置呢？

在 CSS3 中，可以通过 letter-spacing 来设置字符文本之间的距离，这里允许使用负值，这可让字符之间更加紧凑。其语法格式如下：

```
letter-spacing:normal|length
```

其属性值含义如表 10-10 所示。

表 10-10 字符间隔属性值

属性值	说明
normal	默认间隔，即以字符之间的标准间隔显示
length	由浮点数字和单位标识符组成的长度值，允许为负值

【例 10.13】（实例文件：ch10\10.13.html）

```
<!DOCTYPE html>
<html>
<body>
<p style="letter-spacing:normal">Welcome to my home</p>
<p style="letter-spacing:5px">Welcome to my home</p>
<p style="letter-spacing:1ex">这里的字间距是1ex</p>
<p style="letter-spacing:-1ex">这里的字间距是-1ex</p>
<p style="letter-spacing:1em">这里的字间距是1em</p>
```

```
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 10-13 所示，可以看到文字间距以不同大小显示。

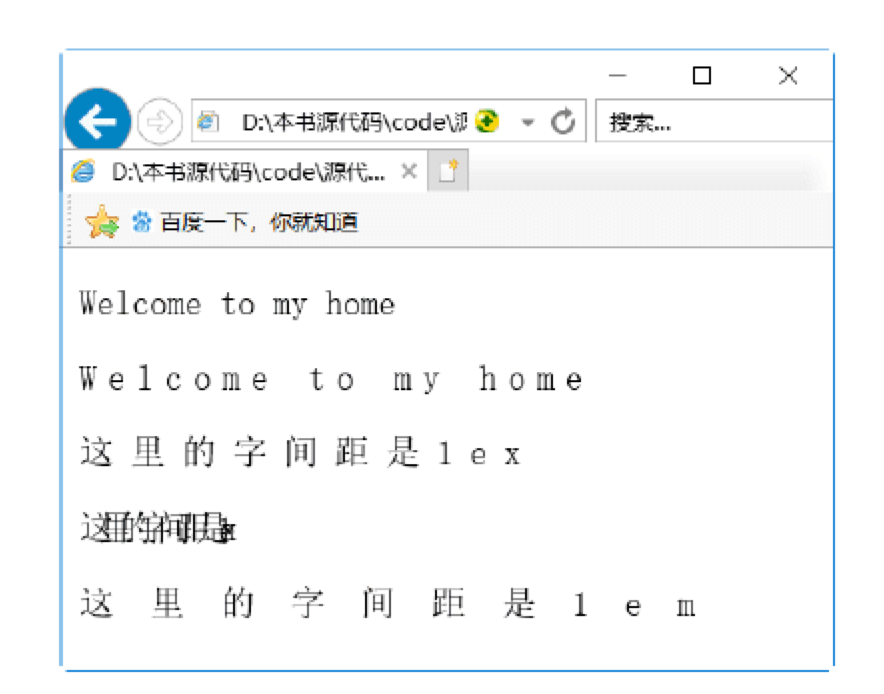


图 10-13　字间距效果



从上述代码中可以看出，通过 letter-spacing 定义了多个字间距的效果。特别注意，当设置的字间距为负值时，所有文字就会粘到一块。

10.3.3　文字修饰 text-decoration

在网页文本编辑中有的文字需要突出重点，告诉读者这段文本很重要，往往会给文字增加下画线，上画线或删除线效果，从而吸引读者的眼球。在 CSS3 中，text-decoration 属性是文本修饰属性，该属性可以为页面提供多种文本的修饰效果，如下画线、删除线、闪烁等。

text-decoration 属性语法格式如下：

```
text-decoration:none|underline|blink|overline|line-through
```

其属性值含义如表 10-11 所示。

表 10-11　文字修饰属性值

属性值	描述
none	默认值，对文本不进行任何修饰
underline	下画线
overline	上画线
line-through	删除线
blink	闪烁

【例 10.14】（实例文件：ch10\10.14.html）

```
<!DOCTYPE html>
```

```
<html>
<body>
  <p style="text-decoration:none">人总是在接近幸福时倍感幸福！</p>
  <p style="text-decoration:underline">人总是在接近幸福时倍感幸福！</p>
  <p style="text-decoration:overline">人总是在接近幸福时倍感幸福！</p>
  <p style="text-decoration:line-through">人总是在接近幸福时倍感幸福！</p>
  <p style="text-decoration:blink">人总是在接近幸福时倍感幸福！</p>
</body>
</html>
```

在 IE 11.0 中显示效果如图 10-14 所示，可以看到段落中出现了下画线、上画线和删除线。

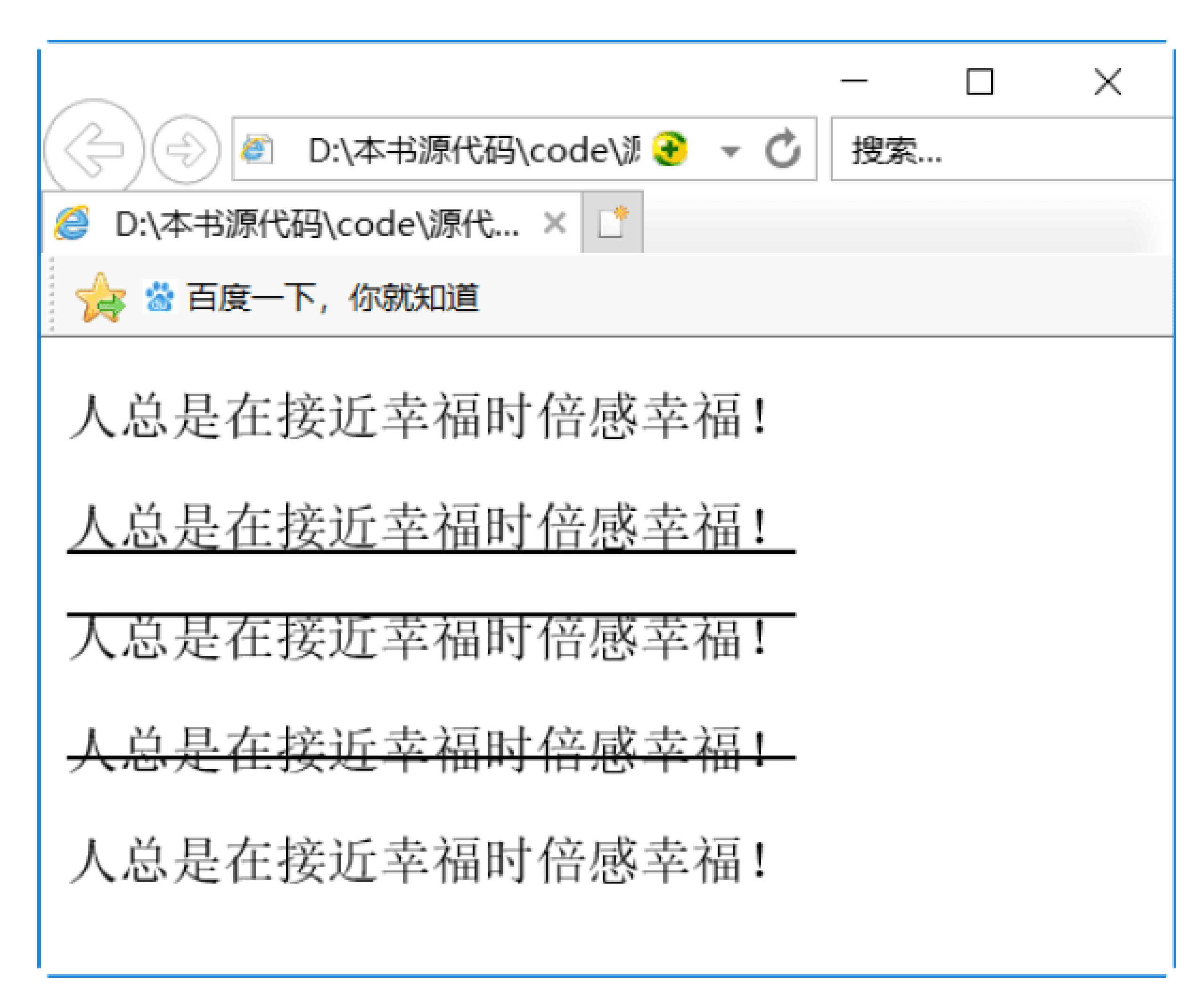


图 10-14　文本修饰显示



这里需要注意的是，blink 闪烁效果只有少数浏览器支持，而 IE 等其他大多数浏览器（如 Opera）都暂不支持该效果。

10.3.4　垂直对齐方式 vertial-align

在网页文本编辑中，对齐有很多方式，文字排在一行的中央位置称为居中对齐，文章的标题和表格中数据一般都居中排列。有时还要求文字垂直对齐，即文字顶部对齐或底部对齐。

在 CSS 中，可以直接使用 vertical-align 属性来定义，该属性用来设置垂直对齐方式。该属性定义行内元素的基线相对于该元素所在行的基线的垂直对齐。允许指定负长度值和百分比值，这会使元素降低而不是升高。在表格中，这个属性可以用来设置单元格内容的对齐方式。

vertical-align 属性语法格式如下：

```
{vertical-align:属性值}
```

vertical-align 属性值如表 10-12 所示。

表 10-12　垂直对齐方式属性值

属性值	说明
baseline	默认，元素放置在父元素的基线上
sub	垂直对齐文本的下标
super	垂直对齐文本的上标
Top	把元素的顶端与行中最高元素的顶端对齐
text-top	把元素的顶端与父元素字体的顶端对齐
middle	把此元素放置在父元素的中部
bottom	把元素的顶端与行中最低的元素的顶端对齐
text-bottom	把元素的底端与父元素字体的底端对齐
length	设置元素的堆叠顺序
%	使用 "line-height" 属性的百分比值来排列此元素。允许使用负值

【例 10.15】（实例文件：ch10\10.15.html）

```
<!DOCTYPE html>
<html>
<body>
<p>世界杯<b style="font-size:8pt;vertical-align:super">2018</b>!
  中国队<b style="font-size:8pt;vertical-align:sub">[注]</b>!
  加油！<img src="1.gif" style="vertical-align:baseline"></p>
<p><img src="2.gif" style="vertical-align:middle"/>
  世界杯！中国队！加油！<img src="1.gif" style="vertical-align:top">
</p><hr>
<p><img src="2.gif" style="vertical-align:middle"/>
世界杯！中国队！加油！<img src="1.gif" style="vertical-align:text-top">
</p>
<p><img src="2.gif" style="vertical-align:middle"/>
世界杯！中国队！加油！<img src="1.gif" style="vertical-align:bottom">
</p>
<hr>
<p><img src="2.gif" style="vertical-align:middle"/>
世界杯！中国队！加油！<img src="1.gif" style="vertical-align:text-bottom">
</p>
<p>
世界杯<b style="font-size:8pt;vertical-align:100%">2018</b>!
中国队<b style="font-size: 8pt;vertical-align:-100%">[注]</b>!
加油！<img src="1.gif" style="vertical-align:baseline">
</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-15 所示，可以看到图文在垂直方向以不同的对齐方式显示。

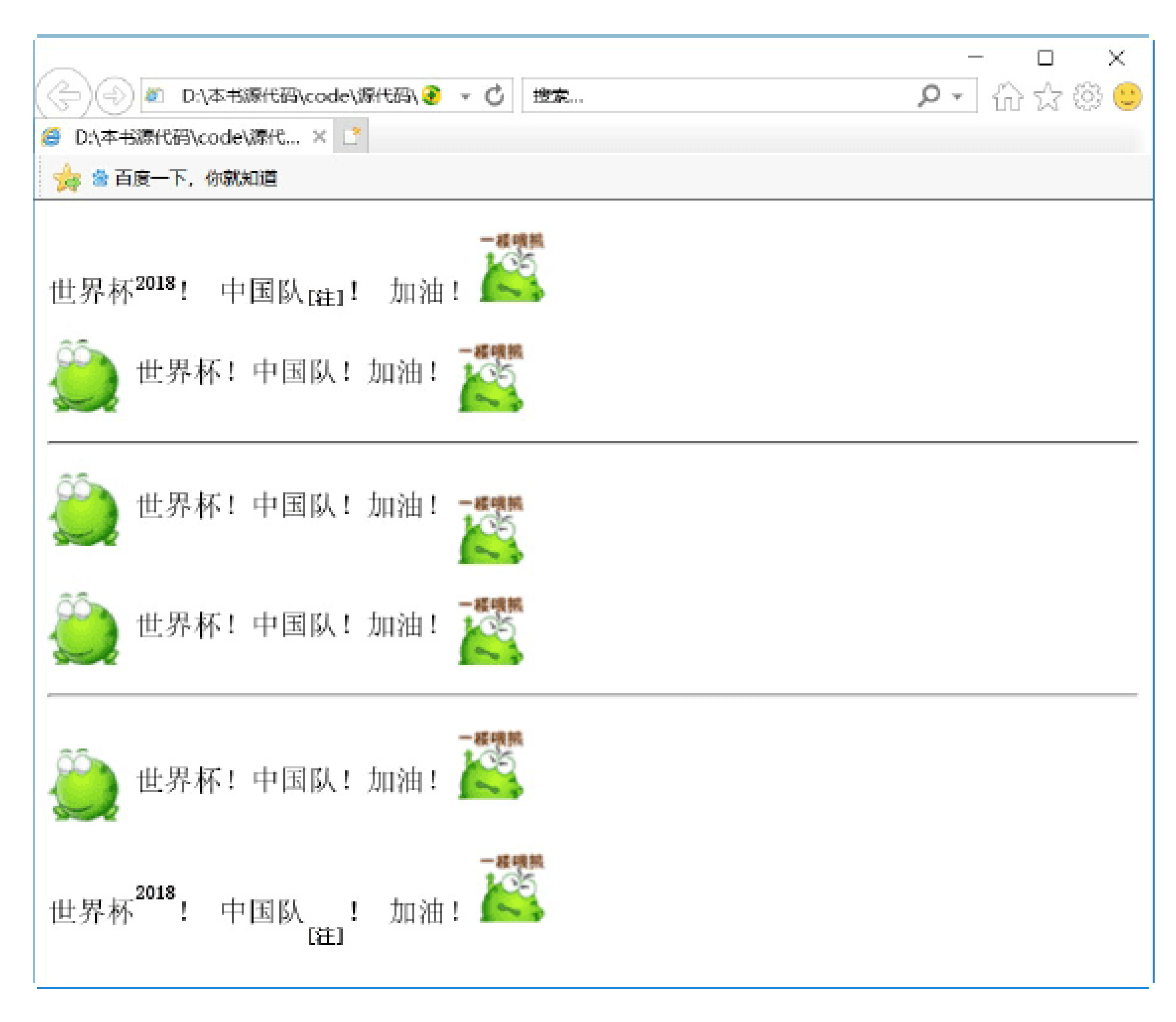


图 10-15　垂直对齐显示

从上面实例中，可以看出上下标在页面中的数学运算或注释标号使用的比较多。顶端对齐有两种参照方式：一种是参照整个文本块；另一种是参照文本。底部对齐同顶端对齐方式相同，分别参照文本块和文本块中包含的文本。



vertical-align 属性值还能使用百分比来设置垂直高度，该高度具有相对性，它是基于行高值来计算的，而且百分比还能使用正负号，正百分比使文本上升，负百分比使文本下降。

10.3.5　文本转换 text-transform

根据需要将小写字母转换为大写字母或者将大写字母转换小写，在文本编辑中都是很常见的。在 CSS 样式中，text-transform 属性可用于设置文本字体的大小写转换。

text-transform 属性语法格式如下：

```
text-transform:none|capitalize|uppercase|lowercase
```

其属性值含义如表 10-13 所示。

表 10-13　文本转换属性值

属性值	说明
none	无转换发生
capitalize	将每个单词的第一个字母转换成大写，其余无转换发生
uppercase	转换成大写
lowercase	转换成小写

因为文本转换属性仅作用于字母型文本，相对来说比较简单。

【例 10.16】（实例文件：ch10\10.16.html）

```
<!DOCTYPE html>
<html>
<body style="font-size:15pt;font-weight:bold">
<p style="text-transform:none">welcome to beijing</p>
<p style="text-transform:capitalize">welcome to beijing</p>
<p style="text-transform:lowercase">WELCOME TO BEIJING</p>
<p style="text-transform:uppercase">welcome to beijing</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-16 所示，可以看到大小写字母转换显示。

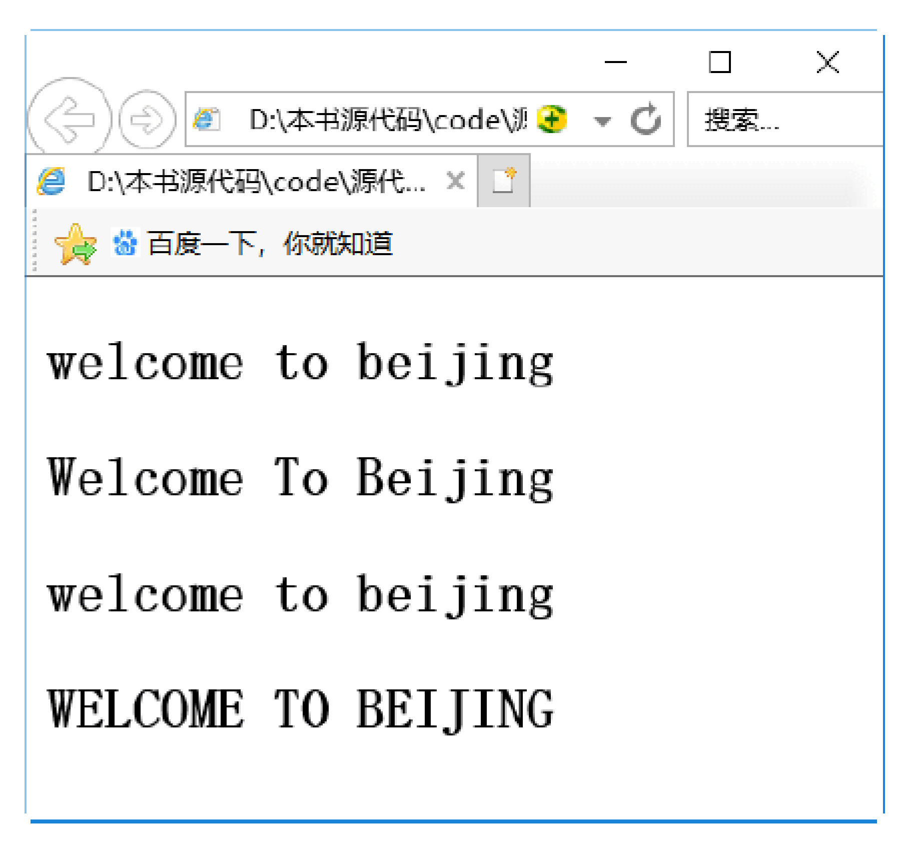


图 10-16　大小写字母转换显示

10.3.6　水平对齐方式 text-align

一般情况下，居中对齐适用于标题类文本，其他对齐方式可以根据页面布局来选择使用。根据需要，可以设置多种对齐，如水平方向上的居中、左对齐、右对齐或两端对齐等。在 CSS 中，可以通过 text-align 属性进行设置。

text-align 属性用于定义对象文本的对齐方式。与 CSS 2 相比，CSS3 增加了 start、end 和 string 属性值。text-align 语法格式如下：

```
{text-align:属性值}
```

其属性值含义如表 10-14 所示。

表 10-14　水平对齐方式属性值

属性值	说明
start	文本向行的开始边缘对齐
end	文本向行的结束边缘对齐
left	文本向行的左边缘对齐。在垂直方向的文本中，文本在 left-to-right 模式下向开始边缘对齐
right	文本向行的右边缘对齐。在垂直方向的文本中，文本在 left-to-right 模式下向结束边缘对齐
center	文本在行内居中对齐
justify	文本根据 text-justify 的属性设置方法分散对齐，即两端对齐，均匀分布
match-parent	继承父元素的对齐方式，但有个例外：继承的 start 或 end 值是根据父元素的 direction 值进行计算的，因此计算的结果可能是 left 或 right
<string>	string 是一个单个的字符，否则就忽略此设置，按指定的字符进行对齐。此属性可以跟其他关键字同时使用，如果没有设置字符，则默认值是 end 方式
inherit	继承父元素的对齐方式

在新增加的属性值中，start 和 end 属性值主要是针对行内元素的（即在包含元素的头部或尾部显示），而<string>属性值主要用于表格单元格中，将根据某个指定的字符对齐。

【例 10.17】（实例文件：ch10\10.17.html）

```
<!DOCTYPE html>
<html>
<body>
<h1 style="text-align:center">登幽州台歌</h1>
<h3 style="text-align:left">选自：</h3>
<h3 style="text-align:right">
<img src="1.gif"/>唐诗三百首</h3>
<p style="text-align:justify">
前不见古人 后不见来者 （这是一个测试，这是一个测试，这是一个测试）
</p>
<p style="text-align:strat">念天地之悠悠</p>
<p style="text-align:end">独怆然而涕下</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-17 所示，可以看到文字在水平方向上以不同的对齐方式显示。

text-align 属性只能用于文本块，而不能直接应用到图像标记<img>。如果要使图像与文本一样应用对齐方式，就必须将图像包含在文本块中。如上例，由于向右对齐方式作用于<h3>

标记定义的文本块，图像包含在文本块中，所以图像能够与文本一样向右对齐。



CSS 只能定义两端对齐方式，但对于具体的两端对齐文本如何分配字体空间以实现文本左右两边均对齐，CSS 并不规定，这就需要设计者自行定义了。



图 10-17 对齐效果

10.3.7 文本缩进 text-indent

在普通段落中，通常首行缩进两个字符，用来表示这是一个段落的开始。同样在网页的文本编辑中可以通过指定属性来控制文本缩进。CSS 的 text-indent 属性可用来设置文本块中首行的缩进。

text-indent 属性语法格式如下：

```
text-indent:length
```

其中，length 属性值表示有百分比数字或有由浮点数字和单位标识符组成的长度值，允许为负值。可以这样认为，text-indent 属性可以定义两种缩进方式：一种是直接定义缩进的长度；另一种是定义缩进百分比。使用该属性，HTML 任何标记都可以让首行以给定的长度或百分比缩进。

【例 10.18】（实例文件：ch10\10.18.html）

```
<!DOCTYPE html>
<html>
<body>
<p style="text-indent:30mm">
此处直接定义长度，直接缩进。
</p>
```

```
<p style="text-indent:10%">
此处使用百分比，进行缩进。
</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-18 所示，可以看到文字以首行缩进方式显示。

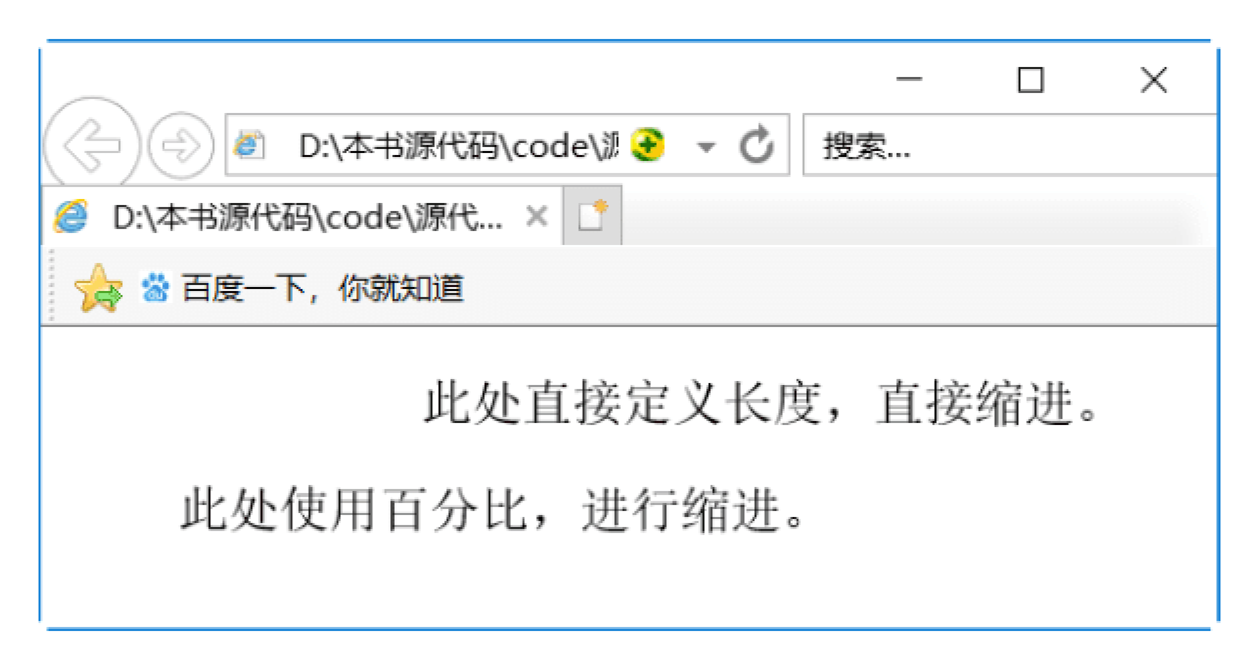


图 10-18　缩进显示窗口

如果上级标记定义了 text-indent 属性，那么子标记可以继承其上级标记的缩进长度。

10.3.8　文本行高 line-height

在 CSS 中，line-height 属性用来设置行间距，即行高。其语法格式如下：

```
line-height:normal|length
```

其属性值的具体含义如表 10-15 所示。

表 10-15　文本行高属性值

属性值	说明
normal	默认行高，即网页文本的标准行高
length	百分比数字或由浮点数字和单位标识符组成的长度值，允许为负值。其百分比取值基于字体的高度尺寸

【例 10.19】（实例文件：ch10\10.19.html）

```
<!DOCTYPE html>
<html>
<body>
<div style="text-indent:10mm;">
  <p style="line-height:50px">
    世界杯（World Cup,FIFA World Cup），国际足联世界杯，世界足球锦标赛)是世界上最
高水平的足球比赛，与奥运会、F1并称为全球三大顶级赛事。
  </p>  <p style="line-height:50%">
     世界杯（World Cup,FIFA World Cup），国际足联世界杯，世界足球锦标赛)是世界上最
```

```
高水平的足球比赛，与奥运会、F1并称为全球三大顶级赛事。
    </p>
  </div>
  </body>
  </html>
```

在 IE 11.0 中浏览效果如图 10-19 所示，可以看到有段文字重叠在一起，此为行高设置过小的结果。



图 10-19　设置文本行高显示效果

10.3.9　处理空白 white-sapce

在网页文本编辑中，有时需要包含一些不必要的制表符、换行符或额外的空白符（多于单词之间的一个标准的空格），这些符号统称为空白字符。通常情况下，浏览器可以自动忽略这些额外的空白字符并按照一种适合窗口的方式布置文本。它会丢弃段落开头和结尾处任何额外的空白，并将单词之间的所有制表符、换行和额外的空白压缩（合并）成单一的空白字符。此外，当用户调整窗口大小，时浏览器会根据需要重新格式化文本以便匹配新的窗口尺寸。对于某些元素，可能会以某种方式特意格式化文本以便包含额外的空白字符，而不抛弃或压缩这些字符。

在 CSS 中，white-space 属性用于设置对象内空格字符的处理方式，该属性对文本的显示有着重要的影响。在标记上应用 white-space 属性可以影响浏览器对字符串或文本间空白的处理方式。

white-space 属性语法格式如下：

```
white-space:normal|pre|nowrap|pre-wrap|pre-line
```

其属性值含义如表 10-16 所示。

表 10-16　处理空白属性值

属性值	说明
normal	默认，空白会被浏览器忽略
pre	空白会被浏览器保留。其行为方式类似 HTML 中的 <pre> 标记
nowrap	文本不会换行，文本会在同一行上继续，直到遇到 标记为止
pre-wrap	保留空白符序列，但是正常地进行换行
pre-line	合并空白符序列，但是保留换行符
inherit	规定应该从父元素继承 white-space 属性的值

【例 10.20】（实例文件：ch10\10.20.html）

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:red; text-align:center;white-space:pre">科    学    发    展观！</h1>
<div>
  <p style="white-space:nowrap;text-indent:10mm">
    重视农业、农村、农民问题是中国共产党一贯的战略思想。<br/>
 “三农”问题始终是关系党和人民事业发展的全局性、根本性问题，农业丰则基础强，农民富则国家盛，农村稳则社会安。”
  </p>
  <p style="white-space:pre-wrap;text-indent:10mm">
     重视农业、农村、农民问题是中国共产党一贯的战略思想。“三农”问题始终是关系党和人民事业发展的全局性、
    根本性问题，农业丰则基础强，<br/>
    农民富则国家盛，农村稳则社会安。
  </p>
  <p style="white-space:pre-line;text-indent:10mm">
       重视农业、农村、农民问题是中国共产党一贯的战略思想。 “三农”问题始终是关系党和人民事业发展的全局性、
    根本性问题，农业丰则基础强，<br/>
    农民富则国家盛，农村稳则社会安。
  </p>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-20 所示，可以看到文字处理空白的不同方式。

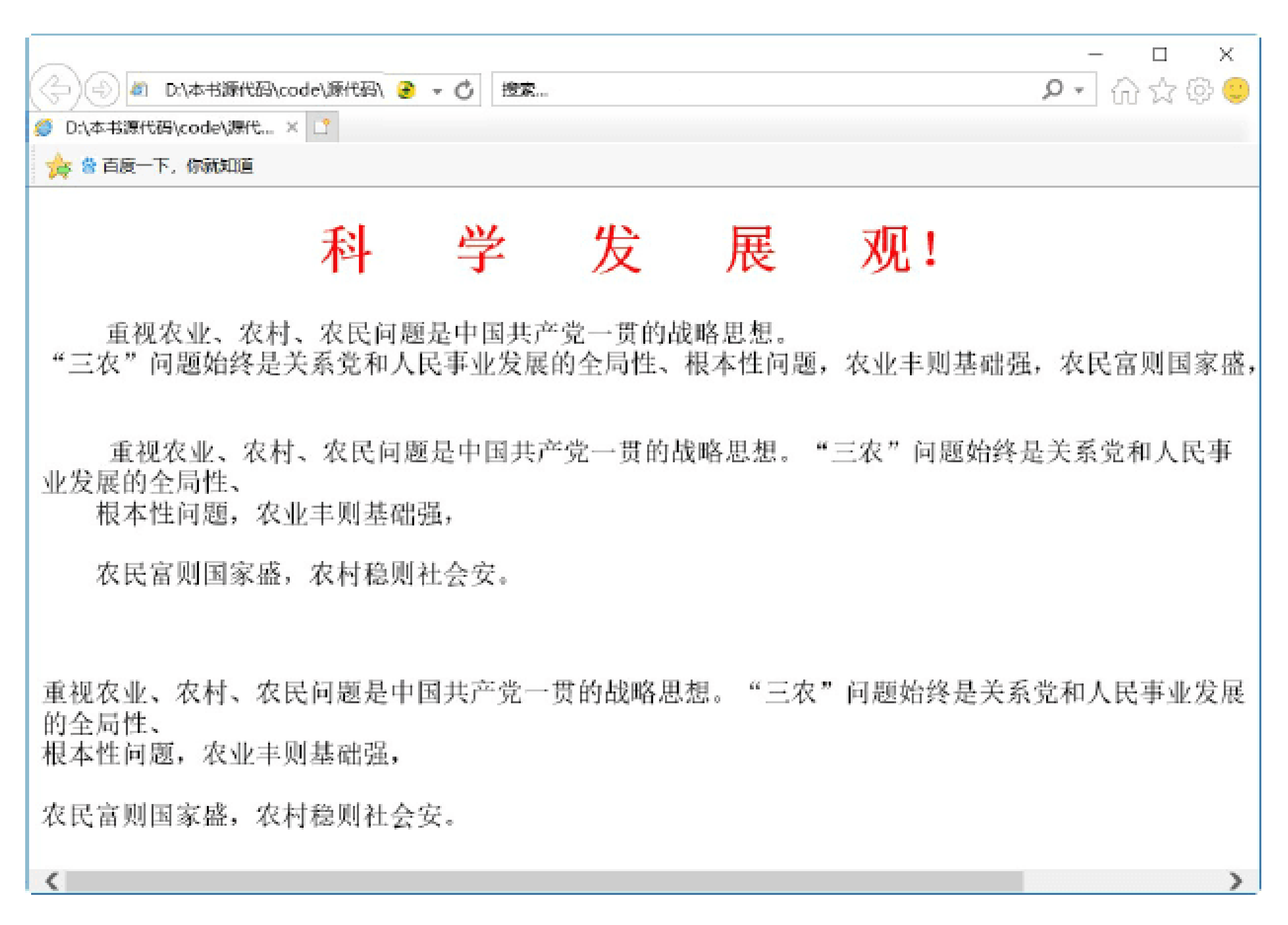


图 10-20　处理空白显示

10.3.10　文本反排 unicode-bidi 和 direction

在网页文本编辑中，通常英语文档的基本方向是从左至右。如果文档中某一段的多个部分包含从右至左阅读的语言，此时可以通过 CSS 提供的 unicode-bidi 和 direction 两个属性解决文本反排的问题。

unicode-bidi 属性语法格式如下：

```
unicode-bidi:normal|bidi-override|embed
```

其属性值含义如表 10-17 所示。

表 10-17　unicode-bidi 属性值

属性值	说明
normal	默认值，元素不会打开一个额外的嵌入级别。对于内联元素，隐式的重新排序将跨元素边界起作用
bidi-override	与 embed 值相同，但除此之外，在元素内，direction 属性将严格按顺序进行排序。此值替代隐式双向算法
embed	元素将打开一个额外的嵌入级别。direction 属性的值指定嵌入级别。重新排序在元素内是隐式进行的

direction 属性用于设置文本流的方向，其语法格式如下：

```
direction:ltr|rtl|inherit
```

其属性值含义如表 10-18 所示。

表 10-18　direction 属性值

属性值	说明
ltr	文本流从左到右
rtl	文本流从右到左
inherit	文本流的值不可继承

【例 10.21】（实例文件：ch10\10.21.html）

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
.a{direction:rtl;unicode-bidi:bidi-override;text-align:left}
</style>
</head>
<body>
<h3>纯CSS反转字符串 使用了direction和unicode-bidi</h3>
<div class=a>秋风吹不尽，总是玉关情。
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 10-21 所示，可以看到文字以反转形式显示。

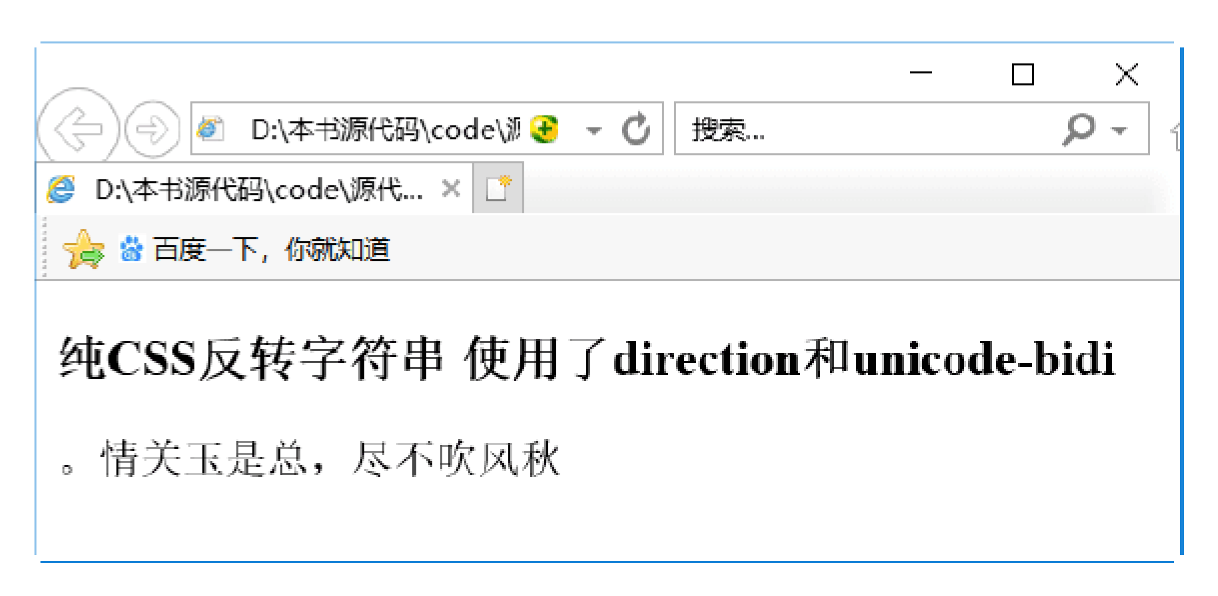


图 10-21　文本反转显示

10.4　综合实例 1——制作旅游宣传网页

在前面小节中，主要介绍了关于文字和段落方面的 CSS 属性设置，在本节中将利用上面的知识创建一个旅游宣传网页，充分利用 CSS 对图片和文字的修饰方法，实现页面效果。具体操作步骤如下：

01 分析需求。

本综合实例要求在网页的最上方显示出标题，标题下方是正文，其中正文部分是图片和文

字段落部分。上述要求使用 CSS 样式属性实现，效果如图 10-22 所示。



图 10-22　旅游宣传网页

02 编写 index.html 文件。

该页面中每个景点的介绍都包括景点图片、景点图片说明及景点介绍，用 HTML5 的 article 表示一个景点，figure 表示景点图片和说明，景点介绍使用段落 p。index.html 文件结构如下：

```
<!DOCTYPE html>
<head>
<meta charset="utf-8" />
<title>河南焦作景点介绍</title>
<link  type="text/css" rel="stylesheet" href="css.css" />
</head>
<body>
<section>
  <h5>景点介绍</h5>
    <article>
      <figure>
          <img src="images/焦作青龙峡3.jpg" width="220" height="140" />
            <figcaption>
              云台山
            </figcaption>
        </figure>
          <p>云台山游览历史悠久，人文景观丰富。据考云台山早在东汉时期就有帝王及其皇室到此采风、避暑；魏晋时不少名士来此避难、隐居；唐宋时受佛教青睐，多处建寺建塔。尤其是唐宋以后，云台山成了文人墨客游山玩水、谈诗论道的主要去处之一。唐代诗人王维曾在此留下了“独在异乡为异客，每逢佳节倍思亲。遥知兄弟登高处，遍插茱萸少一人”的千古绝唱。目前，保留或正在修复的遗迹
```

```
及其他人文景观有东汉黄帝刘协墓——汉献帝陵、“竹林七贤”隐居处——百家岩稠禅师在此建寺、孙思邈炼药处——药王洞、王维做诗处——元贞观，以及万善寺、影寺等。云台山风景游览区于 20 世纪 80 年代初开始经营开发并对外接待游客。 1994 年1月10日被国务院列为国家重点风景名胜区。一年四季游客不断，日接待游客最高达 3.9 万人。经过近二十年的开发和修复 ，游览区内现有老潭沟……</p>
        </article>

        <article>
          <figure>
              <img src="images/焦作影视城.jpg" width="220" height="140" />
                <figcaption>
                  神农山
                </figcaption>
            </figure>
             <p>神农山风景名胜区，是世界地质公园、世界自然基金组织A级优先保护区、国家AAAA级风景旅游区 、国家级猕猴自然保护区、省级科普基地，它位于沁阳市城区西北23公里的太行山麓，共有八大景区136个景点 ，占地总面积为96平方公里。主峰紫金顶海拔1028米，矗立中天，气势雄浑，三大天门比泰山早154年。这里曾是炎帝神农辨百谷、尝百草、登坛祭天的圣地，也是道教创始人老子筑炉炼丹、成道仙升之所。古往今来，优美的自然风光吸引不少帝王将相、文人墨客到此游览，唐明皇李隆基、韩愈、李商隐等历代名家曾在此留下许多传世佳作。这里有雄奇险峻的紫金坛，更有天下一绝的白松岭。15600余株白鹤松姿态万千 、风情万种、婀娜多姿地生长于悬崖绝岭之巅，居世界五大美人松之首。神农山一年四季景色不同，春赏桃花烂漫、夏看流泉飞瀑、秋观满山红叶、冬览冰霜玉龙，游走其间，移步换景，恍若人间仙境，令人魄悸魂动，陡然升华……</p>
        </article>

        <article>
          <figure>
              <img src="images/焦作青龙峡.jpg" width="220" height="140" />
                <figcaption>
                  群英湖
                </figcaption>
            </figure>
             <p>群英湖风景名胜区地处太行山前沿，面积约25平方公里，跨越焦作市区、修武县、博爱县与山西省晋城市地界。群英湖风景区内景点集中，分布均匀。河流、湖泊深秀，高山、峡谷险峻，悬崖、溶洞遍布，奇峰、怪石林立。有寺庙、古树，有台地、草坪，有丛林、花卉及多种野生动植物，还有众多古迹和神话传说，更有世界最高的砌石拱坝这一雄伟的景观，真可谓“群英荟萃”。景区内各类风景自然交织，环境幽静，山清水秀，确是一处难得的旅游胜地。 大坝风光 群英湖坝高 100.5米，是我国最高的砌石坝。大坝耸立于高山峡谷之中，气势雄伟挺拔，造型美观，曾先后以图片的形式在国际大坝会议和广交会上介绍展出。我国正式出版的《中国大坝》、《中国拱坝》图集，以及有关坝工建设的文献资料，都将群英湖大坝作为典型予以刊登。群英湖大坝确实为我国坝工建设上的一枚奇葩。三潭映月景观集线瀑、帘瀑、绿潭于一体，呈“7”梯状分布，天高云淡，四面环山，丛林茂密，溪流不断，是人们假日休闲的好去处……</p>
        </article>

         <article>
          <figure>
              <img src="images/焦作沁阳神农坛风景图4.jpg" width="220" height="140" />
```

```
                <figcaption>
                  青龙峡
                </figcaption>
            </figure>
             <p>焦作青龙峡位于河南省焦作市修武县，是河南云台山世界地质公园主要游览区之一，
也是目前全省唯一的峡谷型省级风景名胜区，被誉为“中原第一峡”。 焦作青龙峡气候独特、山清水秀、
环境优美，是一处天然“氧吧”，是原始生态旅游的绝佳去处。青龙峡是集峰、崖、岭、巅、台、沟、
涧、川、瀑、洞等地貌于一体的自然山水型景区。2000年被确定为河南省风景名胜区，总面积108平方
公里，由青龙峡、净影峡、影寺盆地、双庙、猕猴谷、马头山和大山脑七大游览区组成，主要景点100
多处。 主峰青龙峰海拔高达1323米，站在岭巅，大有“举目四观天下小”之感慨。波澜壮阔的望龙瀑、
神奇独特的倒流泉、妙不可言的七彩潭、堪称一绝的“石上春秋”独具特色的溶洞景观，再加上天然原
始的植物群落，构成了一幅幅极富创意的山水画卷……</p>
        </article>
        <br />
        <br />
        <br />
    </section>
    </body>
    </html>
```

03 编写 css.css 文件。

设置网页中默认文字大小为 13 像素，代码如下：

```
*{font-size:13px;}
```

设置网页的背景颜色为浅绿色，代码如下：

```
/*页面背景颜色*/
body{ background-color:"#ddfcca";}
```

设置 section 区块的属性，代码如下：

```
section{
  width:760px;
  margin:0px auto;  /*实现区块水平居中*/
  padding:0px 20px;
  border: 1px #50ad44 solid;
  }
```

为景点介绍标题设置边距、高度及边框颜色，代码如下：

```
h5{
  margin: 10px 20px;                /*设置外边距的大小*/
  height:23px;                      /*设置标题的高度*/
  border-bottom:3px #50ad44 solid; /*设置下边框的样式*/
  text-indent:2em;                  /*设置文本缩进*/
}
```

为景点照片和说明的父对象 figure 设置相关属性，代码如下：

```
figure{
  padding-right:22px;        /*设置右内边距的大小*/
  display:block;            /*使段落生出行内框*/
  float:left;               /*设置元素向左浮动*/
  width:220px;              /*设置元素的高度*/
}
```

为 article 设置相关属性，代码如下：

```
article{
  border-bottom:1px solid #50ad44;  /*设置底部边框样式*/
  line-height:20px;                 /*设置行高的大小*/
  margin-bottom:10px;               /*设置元素的下外边距*/
}
```

为景点介绍段落 p 设置相关属性，代码如下：

```
p{
  margin:10px 13px;     /*设置外边距的大小*/
  text-indent:2em;      /*设置文本缩进*/
}
```

为景点图片说明 figcaption 设置相关属性，代码如下：

```
figcaption{
  text-align:center;              /*设置段落居中显示*/
  color:#003300;                  /*设置段落的颜色*/
  text-decoration:underline;     /*设置段落下画线效果*/
}
```

为景点图片 img 设置相关属性，代码如下：

```
img{margin-left:10px;}        /*设置外边距的大小*/
```

10.5 综合实例 2——网页简单图文混排

在一个新闻网页中，出现最多的就是文字和图片，图文并茂的文章能够生动地表达新闻主题。本实例将利用前面介绍的文本和段落属性创建一个图片的简单混排，复杂的图片混排会在后面介绍。具体步骤如下：

01 分析需求。

本综合实例要求在网页的最上方显示出标题，标题下方是正文，在正文部分显示图片。上述要求使用 CSS 样式实现，效果如图 10-23 所示。



图 10-23　图文混排显示

02 分析布局并构建 HTML。

首先需要创建一个 HTML 页面，并用 DIV 将页面划分为两个层（网页标题层和正文部分）。

03 导入 CSS 文件。

将 CSS 文件以 link 方式导入到 HTML 页面中。此 CSS 文件定义了该页面的所有样式，其导入代码如下：

```
<link href="10-23.css" rel="stylesheet" type="text/css" />
```

04 完成标题部分。

首先设置网页标题部分，创建一个 div 用来放置标题。其 HTML 代码如下：

```
<div>
<h1>女足世界杯前瞻：巴西女足誓擒澳洲女足</h1>
</div>
```

在 CSS 样式文件中修饰 HTML 元素。其 CSS 代码如下：

```
h1{text-align:center;                /*设置段落居中显示*/
text-shadow:0.1em 2px 6px blue;    /*设置文字阴影效果*/
font-size:18px;                    /*设置文本大小*/
}
```

05 完成正文和图片部分。

下面设置网页正文部分，正文中包含一张图片。其 HTML 代码如下：

```
<div>
<p>周四凌晨，女子世界杯分组赛D组首轮的赛事全面展开，其中南美劲旅巴西女足将在德国门兴格拉德巴赫与澳洲女足进行较量。
</p><p> 巴西足球一向被外界认可，巴西女足上届女子世界杯决赛中遗憾地0比2不敌德国女足，球员满腹怨气，今届欲卷土重来。</p>
<div class="im">
<img src="8.jpg">
</div>
<p>在近6场国际赛中，巴西女足取得4胜2和的不败战绩，在备战今届赛事中的两场热身赛中，相继3比0完胜智利女足和4比1大胜阿根廷女足，显得游刃有余。此番中立场面对实力较弱的澳洲女足，巴西女足有望取得开门红。澳洲女足上届女子世界杯在半准决赛中不敌巴西女足，止步八强，虽然在上场国际赛中主场2比1力克新西兰女足，但在之前2场国际赛分别以1比2相同的比分不敌德国女足和韩国女足，近3场失球多达5个，澳洲女足的后防线漏洞恐怕难以抵挡巴西女足的强大攻势，加上往绩上澳洲女足3战皆负，本场对阵实力强大的巴西女足，澳洲女足只能寄望输少当赢。
</p>
</div>
```

CSS 样式代码如下：

```
p{text-indent:8mm;          /*设置文本缩进*/
line-height:7mm;            /*设置行高的大小*/
}
.im{width:300px;            /*设置图片的宽度*/
float:left;                 /*设置图片向左浮动*/
border:#000000 solid 1px;   /*设置图片的边框样式*/
}
```

10.6 专家解惑

1. 字体为什么在别的电脑上不显示？

楷体很漂亮，草书也不逊色于宋体，但不是所有人的电脑都安装有这些字体，所以在设计网页时，不要为了追求漂亮美观，而采用一些比较新奇的字体，有时这样往往达不到效果。用最基本的字体是最好的选择。

2. 网页中空白需要处理吗？

争取不留空白，即使有足够的空间，也不要用图像、文本或不必要的动画 GIF 来填充网页，在设计时应该避免这种情况发生。

3. 文字和图片导航速度哪个更快？

使用文字做导航栏。文字导航不仅速度快，而且更稳定。例如，有些用户上网时会关闭图片。在处理文本时，不要在普通文本上添加下画线或颜色。除非特别需要，否则不要为普通文字添加下画线，避免使读者误认为文字能够点击。

第11章　CSS3美化表格和表单样式

HTML 数据表格和表单都是网页中常见的元素，表格通常用来显示二维关系数据和排版，从而达到页面整齐和美观的效果。而表单是作为客户端和服务器交流的窗口，可以获取客户端信息，并反馈服务器端信息。本章将介绍使用 CSS3 样式表美化表格和表单样式的方法。

11.1　表格基本样式

在传统网页设计中，表格一直占有比较重要的地位，用来对网页排版布局。基本上每个网页中都有一个表格来布局，它可以帮助设计者控制网页布局、控制内容的显示等。使用表格排版网页可以使网页更美观，条理更清晰，更易于维护和更新。本节将主要介绍如何使用 CSS3 设置表格边框和表格背景色。

11.1.1　表格边框样式

在显示一个表格数据时，通常都带有表格边框，以便用来界定不同单元格的数据。当 table 表格的描述属性 border 值大于 0 时显示边框；如果 border 值为 0，则不显示边框。边框显示之后，可以使用 CSS3 的 border 属性及衍生属性和 border-collapse 属性对边框进行修饰，其中 border 属性表示对边框进行样式、颜色和宽度设置，从而达到提高样式效果的目的，这个属性前面已经介绍过了，其使用方法与前面一模一样，只不过修饰的对象变换了。

border-collapse 属性主要用来设置表格的边框是否被合并为一个单一的边框，还是像在标准的 HTML 中那样分开显示。其语法格式如下：

```
border-collapse:separate|collapse
```

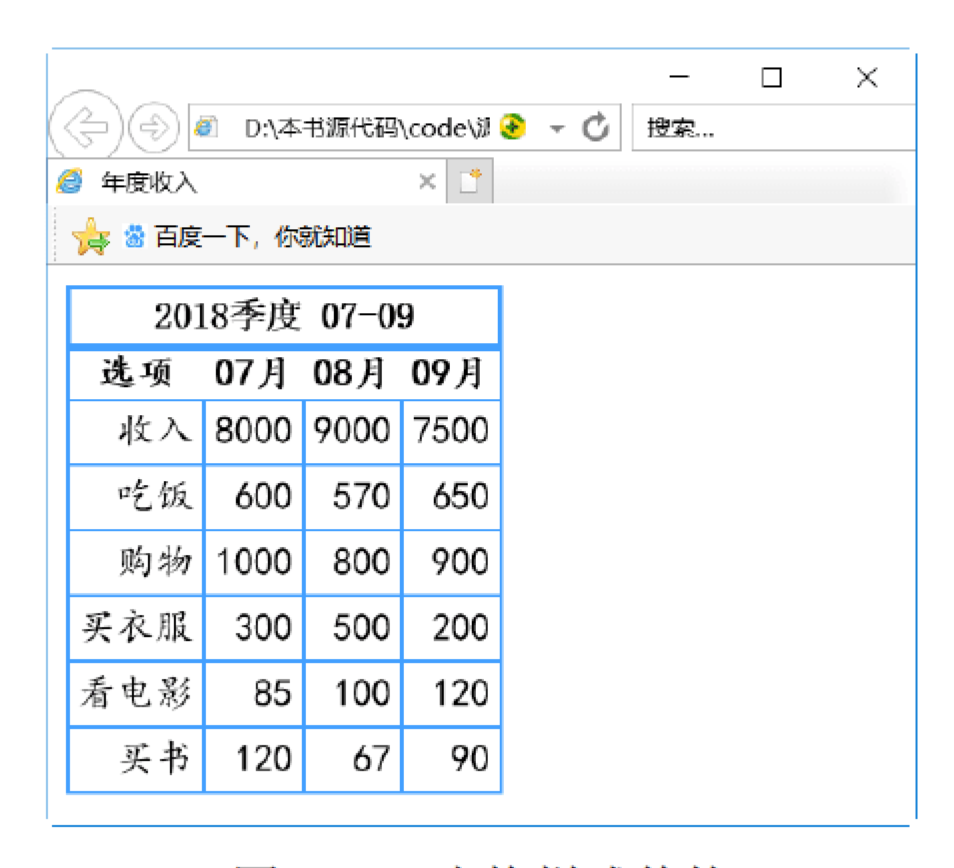
其中 separate 是默认值，表示边框会被分开，不会忽略 border-spacing 和 empty-cells 属性；而 collapse 属性表示边框会合并为一个单一的边框，会忽略 border-spacing 和 empty-cells 属性。

【例 11.1】（实例文件：ch11\11.1.html）

```
<!DOCTYPE html>
<html>
<head>
<title>年度收入</title>
<style>
<!--
.tabelist{
    border:1px solid #429fff;   /* 表格边框 */
    font-family:"楷体";
    border-collapse:collapse;   /* 边框重叠 */
}
.tabelist caption{
    padding-top:3px;          /*设置上内边距的大小*/
    padding-bottom:2px;       /*设置下内边距的大小*/
    font-weight:bolder;      /*设置字体的粗细*/
    font-size:15px;          /*设置字体的大小*/
    font-family:"幼圆";         /* 设置文本的字体 */
    border:2px solid #429fff;   /* 表格标题边框 */
}
.tabelist th{
    font-weight:bold;        /*设置字体的粗细*/
    text-align:center;      /*设置段落居中显示*/
}
.tabelist td{
    border:1px solid #429fff;   /* 单元格边框 */
    text-align:right;         /*设置段落靠右显示*/
    padding:4px;              /*设置内边距的宽度*/
}
-->
</style>
  </head>
<body>
<table class="tabelist">
    <caption class="tabelist">
    2018季度 07-09
    </caption>
    <tr>
      <th>选项</th>
        <th>07月</th>
        <th>08月</th>
        <th>09月</th>
    </tr>
    <tr>
        <td>收入</td>
        <td>8000</td>
        <td>9000</td>
        <td>7500</td>
    </tr>
    <tr>
        <td>吃饭</td>
        <td>600</td>
```

```
            <td>570</td>
            <td>650</td>
        </tr>
        <tr>
            <td>购物</td>
            <td>1000</td>
            <td>800</td>
            <td>900</td>
        </tr>
        <tr>
            <td>买衣服</td>
            <td>300</td>
            <td>500</td>
            <td>200</td>
        </tr>
        <tr>
            <td>看电影</td>
            <td>85</td>
            <td>100</td>
            <td>120</td>
        </tr>
        <tr>
            <td>买书</td>
            <td>120</td>
            <td>67</td>
            <td>90</td>
        </tr>
    </table>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 11-1 所示，可以看到表格带有边框显示，其边框宽度为 1 像素，直线显示并将边框进行合并；表格标题“2018 季度 07-09”也带有边框显示，字体大小为 150 像素，字体类型是幼圆并加粗显示；表格中每个单元格都以 1 像素直线的方式显示边框并将显示对象右对齐。



2018季度 07-09

选项	07月	08月	09月
收入	8000	9000	7500
吃饭	600	570	650
购物	1000	800	900
买衣服	300	500	200
看电影	85	100	120
买书	120	67	90

图 11-1　表格样式修饰

对于上面的例子，我们会发现没有使用 HMTL 标记中的 border 设置边框，而是使用 CSS3 的属性 border 来设置 table 边框，这样做可以在不同浏览器上显示相同样式。

11.1.2 表格边框宽度

虽然使用 HTML 标记的描述 border 也能提高表格的宽度，但我们还是推荐使用 CSS 属性设置边框宽度，如使用 border-width 对边框宽度进行设置。如果需要单独设置某一个边框宽度，则可以使用 border-width 的衍生属性设置，如 border-top-width 和 border-left-width 等。

【例 11.2】（实例文件：ch11\11.2.html）

```
<!DOCTYPE html>
<html>
<head>
<title>表格边框宽度</title>
<style>
table{text-align:center;      /*设置居中显示*/
width:500px;           /*设置表格的宽度*/
border-width:6px;      /*设置边框的宽度*/
border-style:double;  /*设置边框的样式为双线*/
color:blue;           /*设置文本的颜色*/
}
td{border-width:3px;       /*设置边框的宽度*/
border-style:dashed; /*设置边框的样式为虚线*/
}
</style>
</head>
<body>
<table border=1 cellspacing="3" cellpadding="0">
  <tr>
  <td>姓名</td>
  <td class=tds>性别</td>
  <td>年龄</td>
  </tr>
  <tr>
  <td>张三</td>
  <td>男</td>
  <td>31</td>
  </tr>
  <tr>
  <td>李四 </td>
  <td>男</td>
  <td>18</td>
  </tr>
</table>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 11-2 所示，可以看到表格带有边框，宽度为 6 像素，双线式表示，字体颜色为蓝色，单元格边框宽度为 3 像素，显示样式是破折线式。

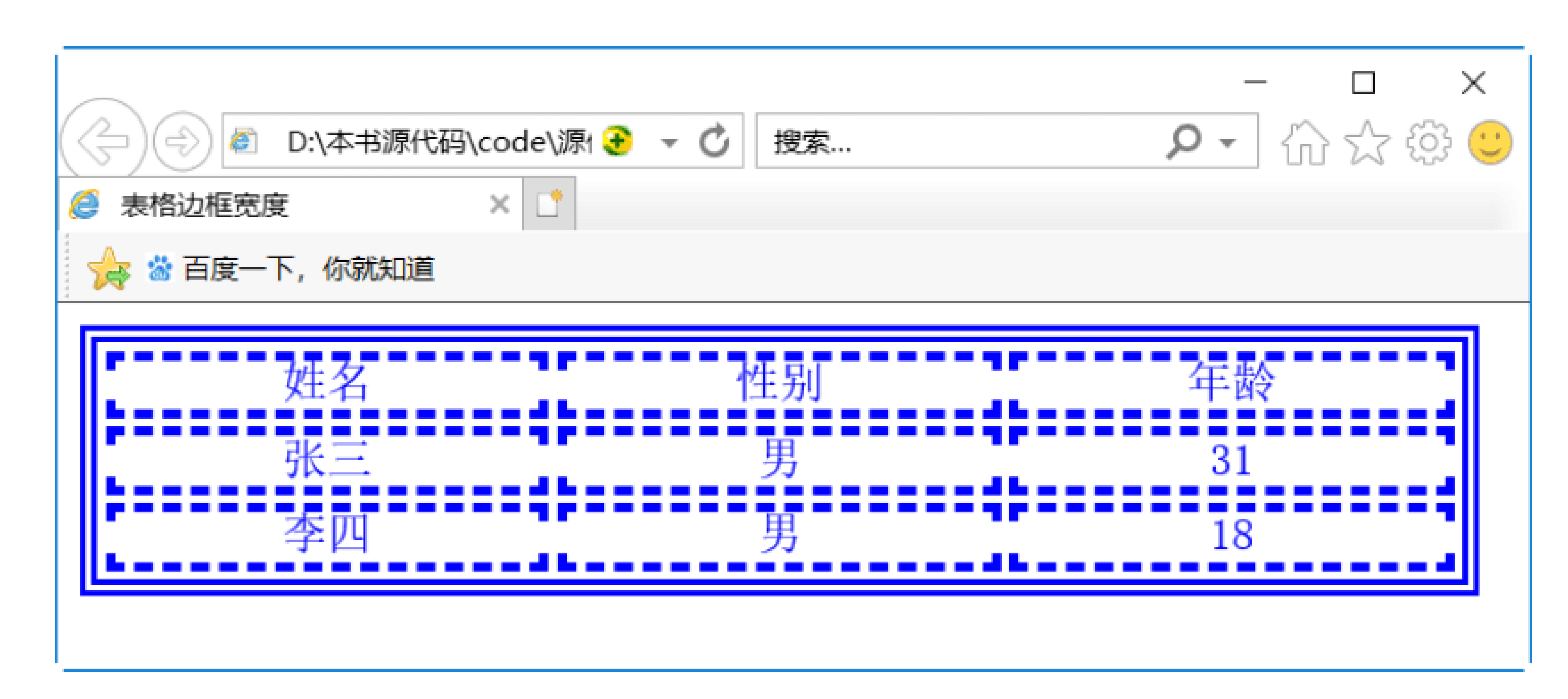


图 11-2　设置表格宽度

11.1.3　表格边框颜色

表格颜色设置非常简单，通常使用 CSS3 属性 color 设置表格中文本颜色，使用 background-color 设置表格背景色。如果为了突出表格中的某一个单元格，还可以使用 background-color 设置某一个单元格的颜色。

【例 11.3】（实例文件：ch11\11.3.html）

```
<!DOCTYPE html>
<html>
<head>
<title>表格边框色和背景色</title>
<style>
*{
    padding:0px;          /*设置内边距的大小*/
    margin:0px;           /*设置外边距的大小*/
}
body{
    font-family:"宋体";  /*设置文本字体样式*/
    font-size:12px;           /*设置字体大小*/
}
table{
    background-color:yellow;  /*设置背景颜色为黄色*/
    text-align:center;        /*设置居中显示*/
   width:500px;             /*设置表格宽度*/
   border:1px solid green;  /*设置表格边框的粗细和颜色*/
}
td{
  border:1px solid green;  /*设置单元格边框的粗细和颜色*/
  height:30px;             /*设置单元格的高度*/
```

```
    line-height:30px;       /*设置行高的大小*/
  }
  .tds{
    background-color:#FFE1FF; /*设置单元格的背景颜色*/
  }
  </style>
  </head>
  <body>
  <table  cellspacing="3" cellpadding="0">
    <tr>
    <td>姓名</td>
    <td class=tds>性别</td>
    <td>年龄</td>
    </tr>
    <tr>
    <td>刘天翼</td>
    <td>男</td>
    <td>32</td>
    </tr>
    <tr>
    <td>刘天佑</td>
    <td>女</td>
    <td>28</td>
    </tr>
  </table>
  </body>
  </html>
```

在 IE 11.0 中浏览效果如图 11-3 所示，可以看到表格带有边框，边框样式显示为绿色，表格背景色为黄色，其中一个单元格背景色为浅紫色。



图 11-3　设置边框背景色

11.2　CSS3 与表单

表单可以用来向 Web 服务器发送数据，特别是经常被用在主页页面——用户输入信息然

后发送到服务器中。实际用在 HTML 中的标记有<form>、<input>、<textarea>、<select>和<option>。本节将使用 CSS3 相关属性对表单进行美化。

11.2.1 美化表单中元素

表单中元素非常多而且杂乱，如 input 输入框、按钮、下拉菜单、单选按钮和复选框等。当使用 form 表单将这些元素排列组合在一起的时候，其单纯的表单效果非常简陋，这时设计者可以通过 CSS3 相关样式控制表单元素输入框、文本框等元素外观。

在网页中，表单元素的背景色默认都是白色的，因为这样的背景色不能美化网页，所以可以使用颜色属性定义表单元素的背景色。定义表单元素背景色可以使用 background-color 属性定义，这样可以使表单元素不那么单调。例如：

```
input{
    background-color:#ADD8E6;
}
```

上面代码设置了 input 表单元素背景色，都是统一的颜色。

【例 11.4】（实例文件：ch11\11.4.html）

```
<!DOCTYPE html>
<html>
<head>
<style>
input{                              /* 所有input标记 */
    color:#cad9ea;
}
input.txt{                          /* 文本框单独设置 */
    border:1px inset #cad9ea;
    background-color:#ADD8E6;
}
input.btn{                          /* 按钮单独设置 */
    color:#00008B;
    background-color:#ADD8E6;
    border:1px outset #cad9ea;
    padding:1px 2px 1px 2px;
}
select{
    width:80px;
    color:#00008B;
    background-color:#ADD8E6;
    border:1px solid #cad9ea;
}
textarea{
    width:200px;
```

```
        height:40px;
        color:#00008B;
        background-color:#ADD8E6;
        border:1px inset #cad9ea;
    }
    </style>
    </head>
    <body>
    <h3>聊天室注册页面</h3>
    <table border="1" width="45%">
    <form method="post">
    <tr><td width="30%">昵称:</td><td><input  class=txt>1－20个字符<div
id="qq"></div></td></tr>
    <tr><td>密码:</td><td><input type="password" >长度为6～16位</td></tr>
    <tr><td>确认密码:</td><td><input type="password" ></td></tr>
    <tr><td>真实姓名: </td><td><input name="username1"></td></tr>
    <tr><td>性别:</td><td><select><option>男</option><option>女
</option></select></td></tr>
    <tr><td>E-mail地址:</td><td><input value="sohu@sohu.com"></td></tr>
    <tr><td>备注:</td><td><textarea cols=35 rows=10></textarea></td></tr>
    <tr><td><input type="button" value="提交" class=btn /></td><td><input
type="reset" value="重填"/></td></tr>
    </form>
    </table>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 11-4 所示，可以看到表单中【昵称】输入框、【性别】下拉框和【备注】文本框都显示了指定的背景颜色。

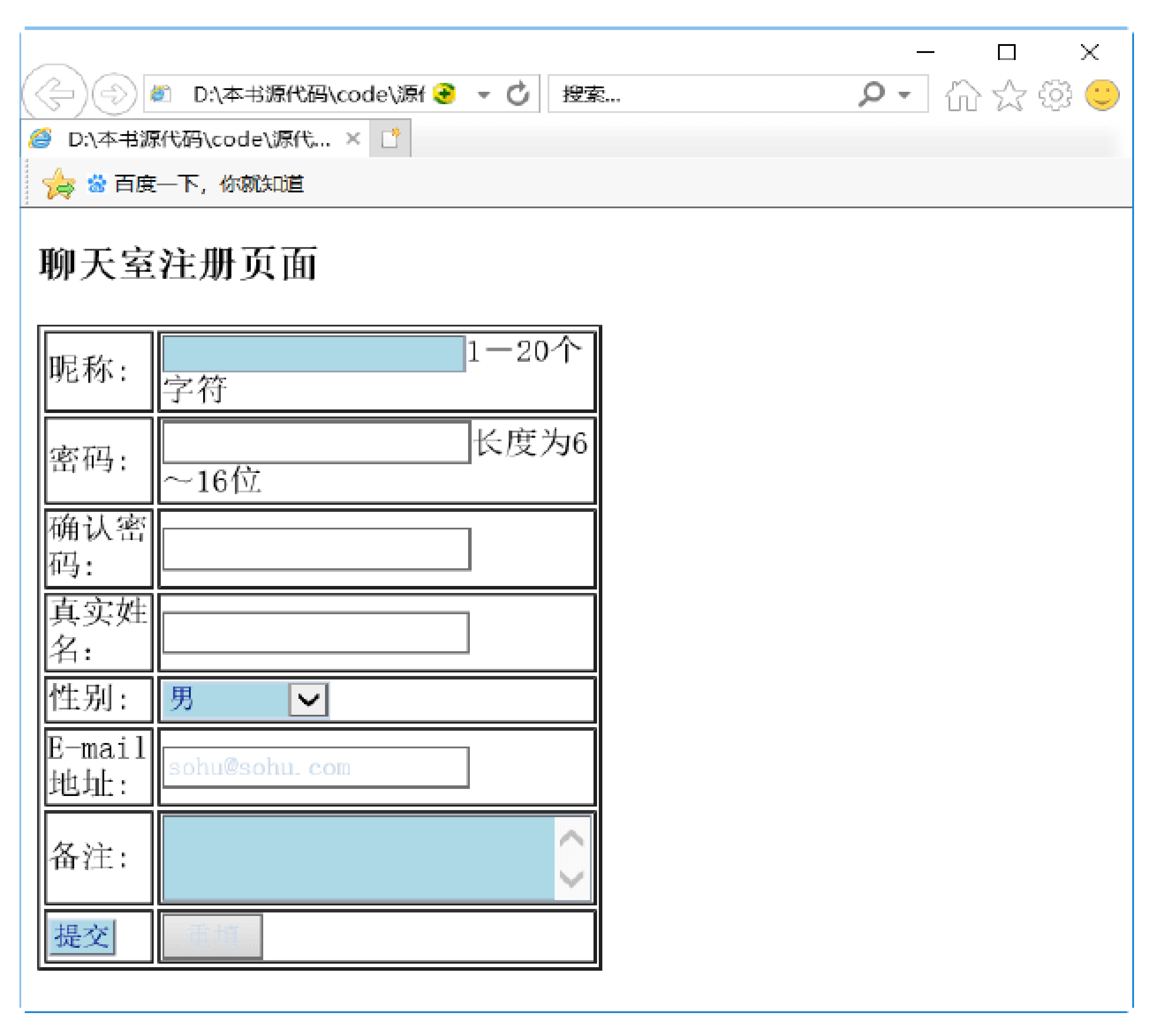


图 11-4　美化表单元素

在上面的代码中，首先使用 input 标记选择符定义了 input 表单元素的字体输入颜色；然后分别定义了两个类（txt 和 btn），txt 用来修饰输入框样式，btn 用来修饰按钮样式；最后分别定义了 select 和 textarean 的样式，其样式定义主要涉及边框和背景色。

11.2.2　美化提交按钮

在网页设计中，还可以使用 CSS 属性来定义表单元素的边框样式，从而改变表单元素的显示效果。例如可以将一个输入框的上、左和右边框去掉，形成一个与签名效果一样的输入框，或者将按钮的 4 个边框去掉，只剩下文字超级链接一样的按钮。

对表单元素边框定义，可以采用 border-style、border-width 和 border-color 及其衍生属性。如果要对表单元素背景色进行设置，就可以使用 background-color，其中将值设置 transparent（透明色）是比较常见的一种方式。示例如下：

```
background-color:transparent;   /* 背景色透明 */
```

【例 11.5】（实例文件：ch11\11.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>表单元素边框设置</title>
<style>
form{
    margin:0px;
padding:0px;
font-size:14px;
}
input{
   font-size:14px;
  font-family:"幼圆";
}
.t{
    border-bottom:1px solid #005aa7;    /* 下画线效果 */
    color:#005aa7;
    border-top:0px; border-left:0px;
    border-right:0px;
    background-color:transparent;       /* 背景色透明 */
}
.n{
    background-color:transparent;       /* 背景色透明 */
    border:0px;                     /* 边框取消 */
}
</style>
</head>
<body>
```

```
<center>
<h1>签名页</h1>
<form method="post">
值班主任: <input id="name" class="t">
<input type="submit" value="提交上一级签名>>" class="n">
</form>
</center>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 11-5 所示，可以看到输入框只剩下一个下边框显示，其他边框被去掉了，提交按钮也只剩下了显示文字，常见的矩形形式被去掉了。



图 11-5 表单元素边框设置

在上面代码中，样式表中定义了两个类标识符 t 和 n，t 用来设置输入框显示样式，此处设置输入框的左、上、下 3 个边框宽度为 0，并设置输入框输入字体颜色为浅蓝色，下边框宽度为 1 像素，直线样式显示，颜色为浅蓝色。在类标识符 n 中，设置背景色为透明色和边框宽度为 0，这样就去掉了按钮常见的矩形样式。

11.2.3 美化下拉菜单

在网页设计中，有时为了突出效果会对文字进行加粗、更换颜色等操作，这样用户就会注意到这些重要文字。同样，也可以对表单元素中的文字进行这样的修饰，下拉菜单是表单元素中常用的元素之一，其样式设置也非常重要。

CSS3 属性不仅可以控制下拉菜单的整体字体和边框，还可以对下拉菜单中的每一个选项设置背景色和字体颜色。对于字体设置可以使用 font 相关属性，如 font-size、font-weight 等；对于颜色设置可以采用 color 和 background-color 属性。

【例 11.6】（实例文件：ch11\11.6.html）

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>美化下来菜单</title>
<style>
.blue{
    background-color:#7598FB;
    color: #000000;
   font-size:15px;
   font-weight:bolder;
   font-family:"幼圆";
}
.red{
    background-color:#E20A0A;
    color: #ffffff;
   font-size:15px;
   font-weight:bolder;
   font-family:"幼圆";
}
.yellow{
    background-color:#FFFF6F;
    color: #000000;
   font-size:15px;
   font-weight:bolder;
   font-family:"幼圆";
}
.orange{
    background-color:orange;
    color:#000000;
   font-size:15px;
   font-weight:bolder;
   font-family:"幼圆";
}
</style>
</head>
<body>
<form method="post">
    <p><label for="color">选择暴雨预警信号级别:</label>
    <select name="color" id="color">
        <option value="">请选择</option>
        <option value="blue" class="blue">暴雨蓝色预警信号</option>
        <option value="yellow" class="yellow">暴雨黄色预警信号</option>
        <option value="orange" class="orange">暴雨橙色预警信号</option>
             <option value="red" class="red">暴雨红色预警信号</option>
    </select></p>
    <p><input type="submit" value="提交"></p>
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 11-6 所示，可以看到下拉菜单中每个菜单项显示不同的背景色。

这种方式显示选项会提高人的注意力，减少犯错的机会。

在上面代码中设置了 4 个类标识符，用来对应不同的菜单选项。其中每个类中都设置了选项的背景色、字体颜色、大小和字体。



图 11-6　设置下拉菜单样式

11.3　综合实例 1——隔行变色

当使用表格显示大量数据的时候，由于表格的行和列比较多，如果单元格采用相同的背景色，用户在查看数据时就会感到非常凌乱，很容易出错。通常的解决办法就是采用隔行变色，使得奇数行和偶数行的背景色不同，从而达到数据一目了然的效果。本节将结合前面学习的知识，创建一个隔行变色实例，具体操作步骤如下：

01 分析需求。

如果要实现表格隔行变色，首先需要创建一个表格并定义其显示样式，然后设置其奇数行和偶然行显示的颜色。

02 创建 HTML 页面，实现基本 table 表格。

```
<!DOCTYPE html>
<html>
<head>
<title>隔行变色</title>
</head>
<body>
<h1>设计优雅数据表格</h1>
<table border=1>
<tr>
<th>排名</th>
<th>姓名</th>
```

```
<th>总分</th>
<th>语文</th>
<th>数学</th>
</tr>
<tr><td>1</td><td>孔　宇</td><td>180</td><td>91</td><td>89</td></tr>
<tr><td>2</td><td>曹圆新</td><td>176</td><td>76</td><td>100</td></tr>
<tr><td>3</td><td>史雅琪</td><td>168</td><td>83</td><td>85</td></tr>
<tr><td>4</td><td>曹秀英</td><td>153</td><td>73</td><td>80</td></tr>
<tr><td>5</td><td>杨　青</td><td>146</td><td>70</td><td>76</td></tr>
</table>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 11-7 所示，可以看到页面中显示了一个表格，其字体、边框等都是默认设置。

03 添加 CSS 代码，设置标题和表格基本样式。

```
<style>
h1{font-size:16px;}           /*设置标题字体大小*/
table{
    width:100%;
    font-size:12px;            /*设置表格字体大小*/
    table-layout:fixed;        /*设置表格布局样式*/
    empty-cells:show;          /*绘制单元格的边框*/
    border-collapse:collapse;  /*边框合并*/
    margin:0 auto;             /*设置外边框的样式*/
    border:1px solid #cad9ea;  /*设置边框的边框粗细和颜色*/
    color:#666;                /*设置字体颜色*/
}
</style>
```

此样式中，设置了标题字体大小为 16 像素，表格字体大小为 12 像素，边框合并，边框大小为 1 像素直线显示等。其中 empty-cells 属性设置是否显示表格中的空单元格（仅用于“分离边框”模式），show 表示显示，hidden 表示隐藏。table-layout:fixed 语句表示表格的水平布局是仅仅基于表格的宽度，表格边框的宽度、单元格间距、列的宽度均与表格内容无关。

在 IE 11.0 中浏览效果如图 11-8 所示，可以看到页面中显示了一个表格，其大小充满了整个页面，边框大小显示为浅蓝色。

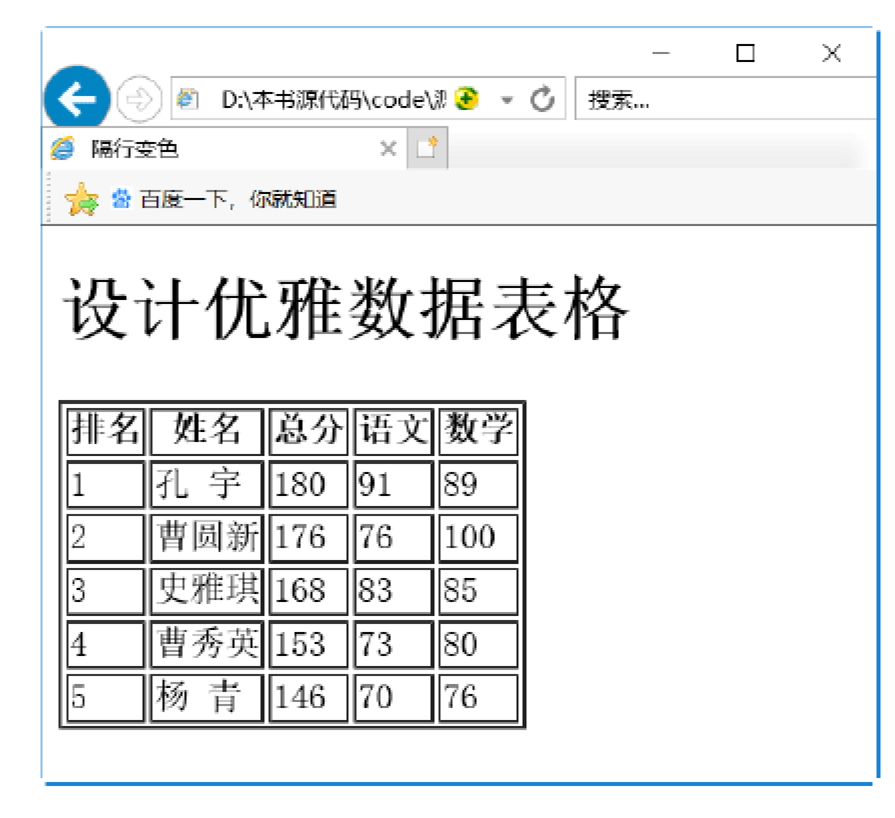


图 11-7　设置 table 表格

图 11-8　设置表格和标题样式

01 添加 CSS 代码，修饰 td 和 th 单元格。

```
th{
   height:30px;
   overflow:hidden;
}
td{height:20px;}
td,th{
   border:1px solid #cad9ea;
   padding:0 1em 0;
}
```

在 IE 11.0 中浏览效果如图 11-9 所示，可以看到表格中单元格高度加大，td 增加到 20 像素，th 增加到 30 像素。单元格还带有边框显示，大小为 1 像素，直线样式，颜色为浅蓝色。

05 添加 CSS 代码，实现隔行变色。

```
tr:nth-child(even){
     background-color:#FFD39B;
}
```

在这里使用了结构伪类标识符实现了表格的隔行变色。

在 IE 11.0 中浏览效果如图 11-10 所示，可以看到表格中奇数行显示浅蓝色。

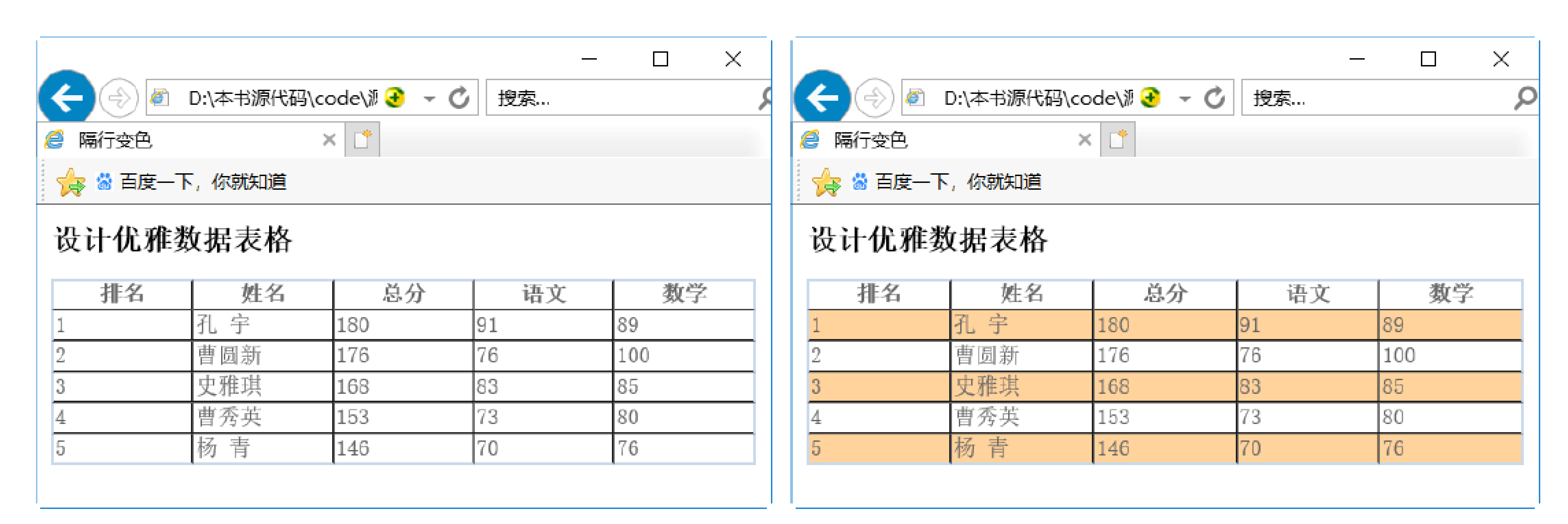


图 11-9　设置单元格样式　　图 11-10　隔行变色显示

11.4 综合实例 2——表格图文网页布局

文字和图片等对象的定位是网页设计中的难点，利用表格布局网页结构是一种非常重要，也是最为常用的方法，同时也是网页设计的重点。通过单元格可以对文本或图片进行定位，以达到布局整齐。本实例将结合前面学习的知识，实现通过表格对图片和文字进行布局，具体操作步骤如下：

01 分析需求。

使用表格进行排版，需要先确定是图文竖排还是横排，这样才能确定图片的放置位置（如果采用竖排，则可以在表格的第一行放置图片，对应的下面放置文字介绍），然后使用 CSS 样式对图片、文字和表格进行设置。

02 创建 HTML 页面，实现表格基本样式。

```
<!DOCTYPE html>
<html>
<head>
<title>无标题文档</title>
</head>
<body>
<table align=center border="1"  cellspacing="4" cellpadding="0">
  <tr>
  <td><img src="8.jpg"/><p>时尚大玩具！车展前实拍路虎极光Coupe
</p>
</td>
  <td><img src="6.jpg"/><p>充满活力的小家伙 上海车展体验奥迪Q3
</p></td>
  </tr>
  <tr>
  <td>路虎总是以硬汉的形象出现在人们面前</td>
  <td>奥迪将推出入门级SUV车型奥迪Q3时，大家都自然而然的觉得这款车型是和宝马X1竞争的
产品</td>
  </tr>
  <tr>
  <td> 全景天窗已经成为中高端SUV的流行趋势</td>
  <td>奥迪Q3相比两位兄长有简化也有复杂化 </td>
  </tr>
</table>
</body>
</html>
```

上面网页代码创建了一个表格，表格第一行分别放置两种图片，下面两行放置文字。

在 IE 11.0 中浏览效果如图 11-11 所示，可以看到显示了一个表格，表格中包含了两张图片。

03 添加 CSS 代码，修饰全局样式和表格。

```
<style>
* {
    padding:0px;                  /*设置内边距的大小*/
    margin:0px;                   /*设置外边距的大小*/

}
table{
    font-family:"幼圆";           /*设置文本的字体样式*/
    text-align:center;           /*设置居中显示*/
margin:10px 0 0 30px;        /*设置表格的外边距*/
    background-color:#CCddCC;  /*设置表格的背景颜色*/
border-color:#0099ff;       /*设置表格的边框颜色*/
width:400px;                /*设置表格的宽度*/
font-size:15px;            /*设置文字的大小*/
}
</style>
```

通过上面的代码即可创建一个 table 标记选择器，该选择器设置了字体样式和背景样式。

在 IE 11.0 中浏览效果如图 11-12 所示，可以看到显示了一个表格，其背景色为绿色，字体居中对齐，表格边框为浅蓝色。



图 11-11 创建基本 table

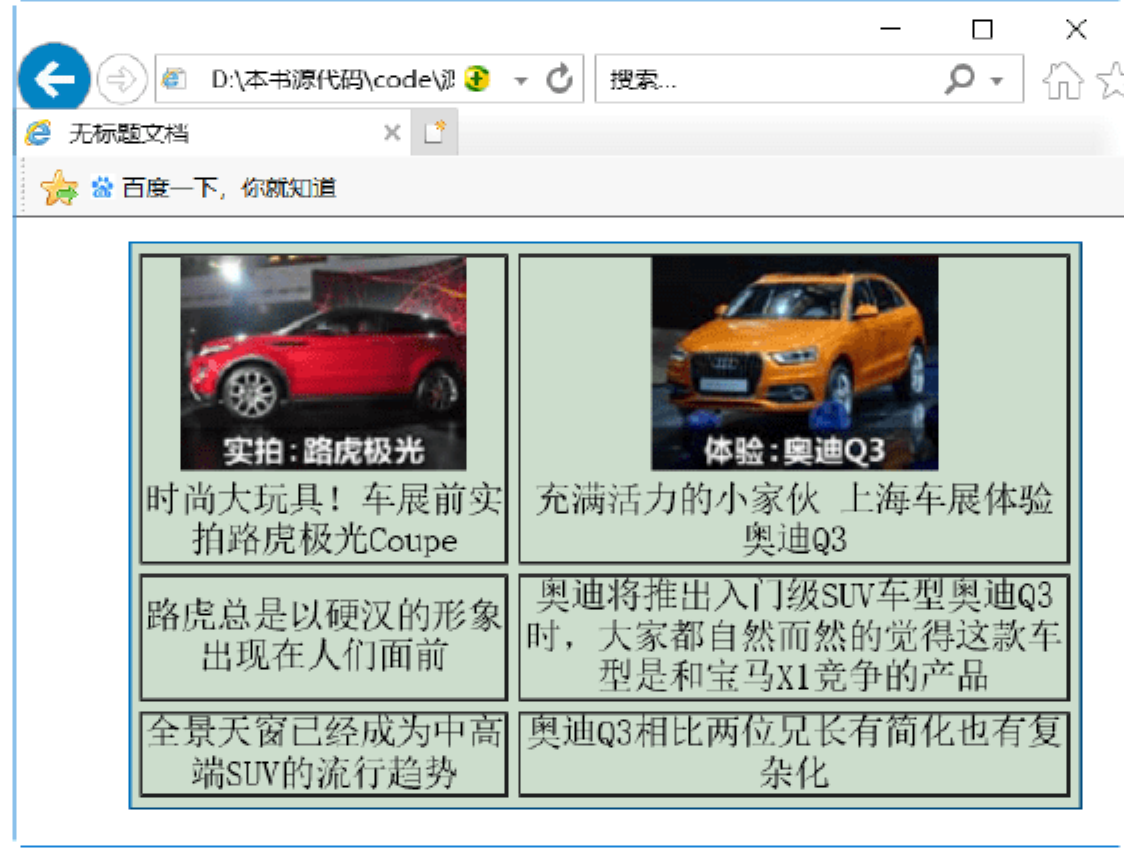


图 11-12 CSS 修饰 table

04 添加 CSS 代码，修饰行 tr 和单元格 td。

```
tr{
    height:30px;        /*设置表格的行高*/
    line-height:30px;   /*设置文字行高*/
}
td{
    width:200px;
    }
```

上面代码首先定义了 table 行的高度和文字行高，然后定义了单元格宽度。

在 IE 11.0 中浏览效果如图 11-13 所示，可以看到显示表格相比较于上一个图像，其表格高度增加，行高增加。

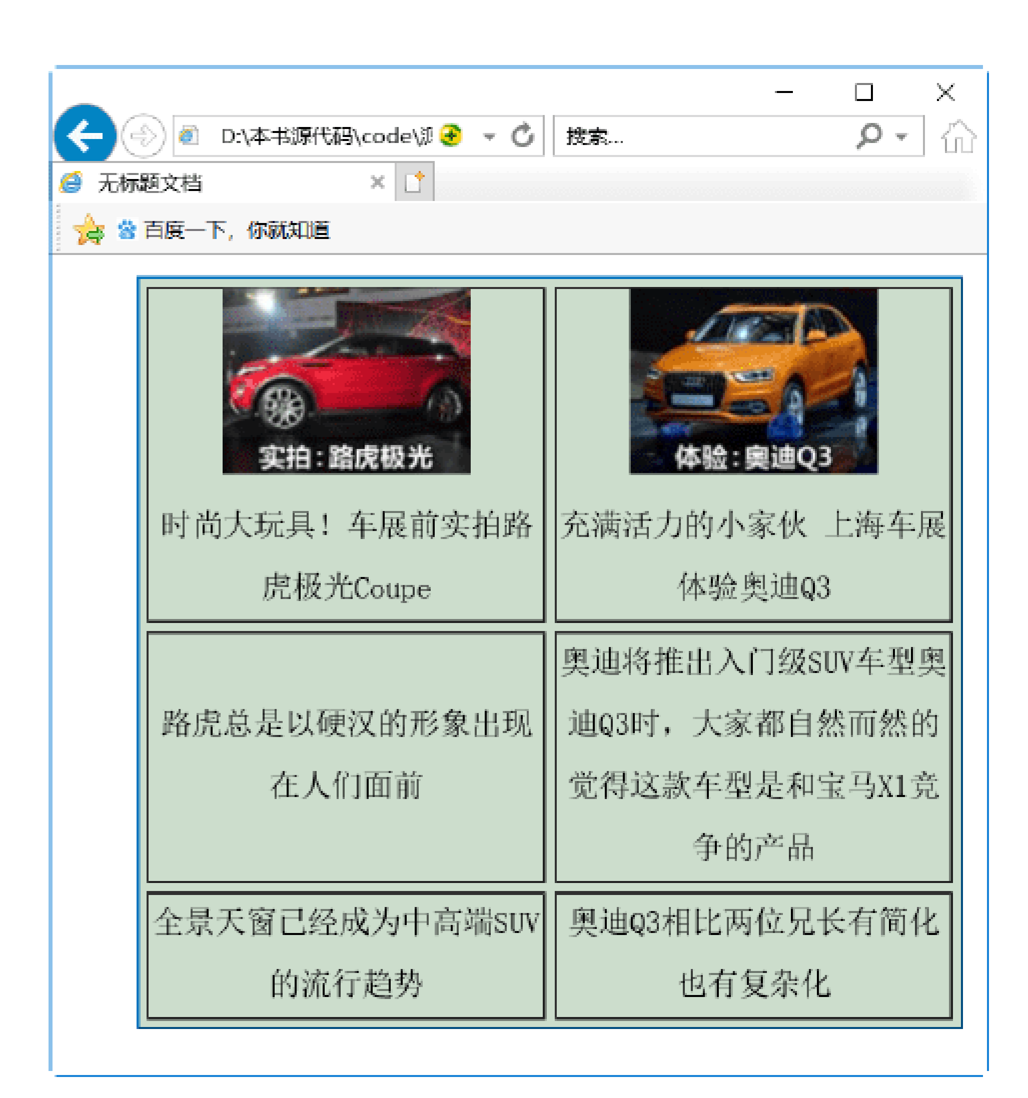


图 11-13　设置行高

11.5　综合实例 3——变色表格

如果长时间浏览大量数据和浏览表格，即使使用了隔行变色的表格，阅读时间长了仍然会感到疲劳。如果数据行能动态根据鼠标悬浮来改变颜色，就会使页面充满动态效果，缓解用户的疲劳感。

本实例将结合前面学习的知识，创建一个鼠标悬浮的变色表格，具体操作步骤如下：

01 分析需求。

首先需要建立一个表格，所有行的颜色不单独设置，统一采用表格本身的背景色，然后通过 CSS 设置可以实现该效果。

02 创建 HTML 网页，实现 table 表格。

```
<!DOCTYPE html>
<html>
<head>
<title>变色表格</title>
</head>
<body>
<table border="0" cellpadding="0" cellspacing="1">
```

```
<caption>
唐宋八大家
</caption>
  <tr>
   <th>姓名</th>
   <th>作品</th>
  </tr>
  <tr class="hui">
    <td>韩愈</td>
    <td>原道</td>
  </tr>
  <tr>
    <td>柳宗元</td>
    <td>三戒</td>
  </tr>
  <tr class="hui">
    <td>欧阳修</td>
    <td>醉翁亭记</td>
  </tr>
  <tr>
    <td>苏洵</td>
    <td>六国论</td>
  </tr>
  <tr class="hui">
    <td>苏轼</td>
    <td>水调歌头</td>
  </tr>
  <tr>
    <td>苏辙</td>
    <td>栾城集</td>
  </tr>
  <tr class="hui">
    <td>曾巩</td>
    <td>上欧阳舍人书</td>
  </tr>
  <tr>
    <td >王安石</td>
    <td>伤仲永</td>
  </tr>
</table>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 11-14 所示，可以看到表格不带边框，字体等都是默认显示。

03 添加 CSS 代码，修饰 table 表格和单元格。

```
<style type="text/css">
```

```
table{
    width:600px;
    margin-top:0px;
    margin-right:auto;
    margin-bottom:0px;
    margin-left:auto;
    text-align:center;
    background-color:#000000;
    font-size:9pt;
}
td{
    padding:5px;
    background-color:#FFFFFF;
}
</style>
```

在 IE 11.0 中浏览效果如图 11-15 所示，可以看到表格带有边框，行内字体居中显示，但列标题背景色为黑色，其中字体不能够显示。

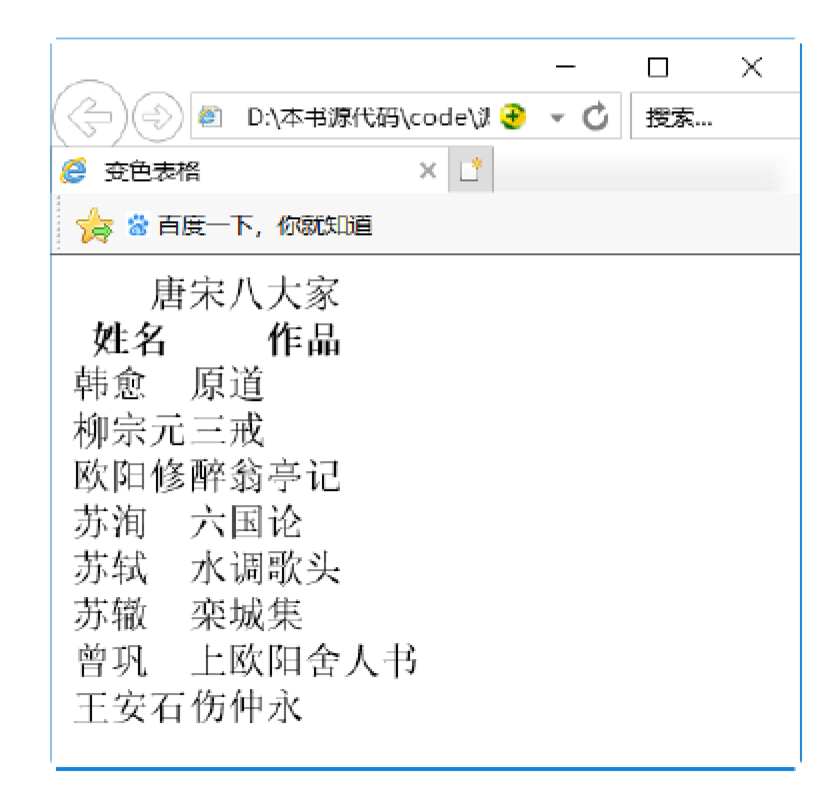


图 11-14 创建基本表格



图 11-15 设置 table 样式

04 添加 CSS 代码，修饰标题。

```
caption{
    font-size:36px;
    font-family:"黑体","宋体";
    padding-bottom:15px;
}
tr{
    font-size:13px;
    background-color:#cad9ea;
    color:#000000;
}
th{
    padding:5px;
}
.hui td{
```

```
        background-color:#f5fafe;
    }
```

上面代码中使用了类选择器 hui，来定义每个 td 行所显示的背景色，此时需要在表格中每个奇数行都引入该类选择器。例如<tr class="hui">从而设置奇数行背景色，这与第一个综合实例中对奇数行背景色的设置方式是不一样的。

在 IE 11.0 中浏览效果如图 11-16 所示，可以看到一个表格中列标题一行背景色显示为浅蓝色，并且表格中奇数行背景色为浅灰色，而偶数行背景色显示为默认白色。

04 添加 CSS 代码，实现鼠标悬浮变色。

```
    tr:hover td{
        background-color:#FF9900;
    }
```

在 IE 11.0 中浏览效果如图 11-17 所示，可以看到当鼠标放到不同行上时背景色会发生改变。

图 11-16　设置奇数行背景色



图 11-17　鼠标悬浮改变颜色

11.6　综合实例 4——登录表单

在网页设计时，构建一个登录表单是十分常见的事情。登录表单在验证用户时扮演着重要的角色，可以保护一些敏感数据不被非法用户访问。设计和构建登录表单时，尽量使其简单，并且避免将一个表单用于多个任务。

本实例将结合前面学习的知识，创建一个简单的登录表单，具体操作步骤如下：

01 分析需求。

创建一个登录表单，需要包含三个表单元素：一个名称输入框、一个密码输入框和两个按钮，然后添加一些 CSS 代码对表单元素进行修饰。

02 创建 HTML 网页，实现表单。

```
<!DOCTYPE html>
<html>
<head>
<title>用户登录</title>
<body>
<div>
<h1>用户登录</h1>
<form action="" method="post">
姓名：<input type="text" id=name/>
密码：<input type="password" id=password/>
<input type=submit value="提交" class=button>
<input type=reset value="重置" class=button>
</form>
</div>
</body>
</html>
```

在上面代码中创建了一个 div 层，用来包含表单及其元素。

在 IE 11.0 中浏览效果如图 11-18 所示，可以看到显示了一个表单，其中包含两个输入框和两个按钮，输入框用来获取姓名和密码，按钮分别为【提交】和【重置】。

03 添加 CSS 代码，修饰标题、层、输入框和按钮。

```
<style>
h1{
       font-size:20px;
   }
div{
     width:200px;
     padding:1em 2em 0 2em;
     font-size:12px;
}
#name,#password{
     border:1px solid #ccc;
     width:160px;
     height:22px;
     padding-left:20px;
     margin:6px 0;
     line-height:20px;
}
.button{margin:6px 0;}
</style>
```

在 IE 11.0 中浏览效果如图 11-19 所示，可以看到标题变小，并且密码输入框换行显示，布

局比原来更加美观合理。

图 11-18　创建登录表单

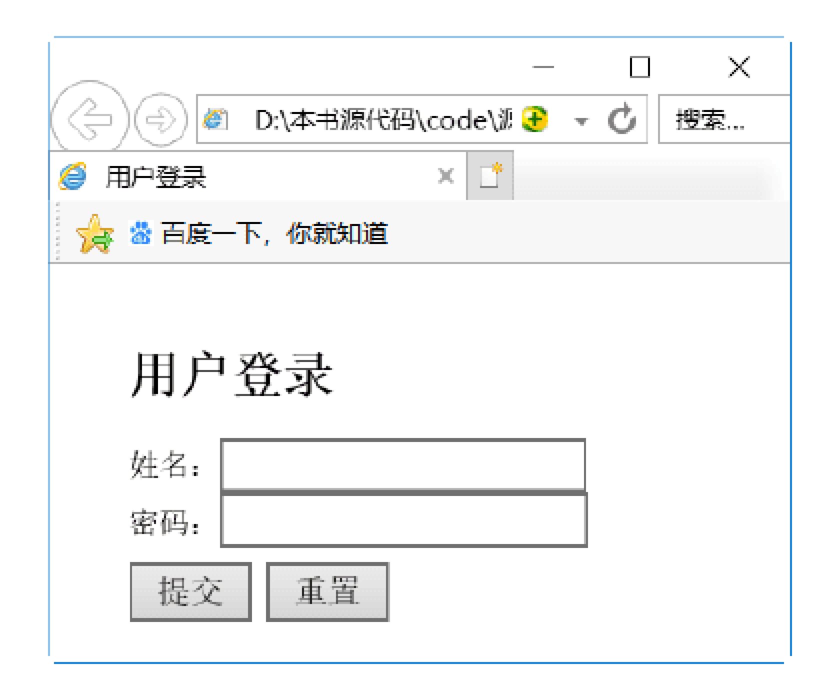


图 11-19　登录表单最终效果

上面代码中，设置了标题大小为 20 像素，div 层宽度为 200 像素，层中字体大小为 12 像素，输入框长度变短，输入框边框变小，并且表单元素之间距离增大，页面布局更加合理。

11.7　综合实例 5——注册表单

不管是在线交易验证、评论新文章，还是管理某个应用，Web 表单总会出现在人们的视线中。在网页上，Web 表单把用户、信息、Web 产品或服务连接了起来，它们能促进销售、捕捉用户行为、建立沟通与交流。

本实例中，将使用表单内的各种元素来开发一个网站的注册页面，并用 CSS 样式来美化这个页面效果。具体操作步骤如下：

01 分析需求。

注册表单通常包含三个部分：在页面上方给出标题；标题下方是正文部分（即表单元素）；最下方是表单元素提交按钮。在设计这个页面时，需要把“用户注册”标题设置成 h1 大小，正文使用 p 来限制表单元素。

02 构建 HTML 页面，实现基本表单。

```
<!DOCTYPE html>
<html>
<head>
<title>注册页面</title>
</head>
<body>
<h1 align=center>用户注册</h1>
<form method="post">
<p>姓    名:
```

```
<input type="text" class=txt size="12" maxlength="20" name="username"/>
</p><p>性    别:
<input type="radio" name="性别" value="male"/>男
<input type="radio" name="性别" value="female"/>女
</p><p>年    龄:
<input type="text" class=txt name="age"/>
</p>
<p>联系电话:
<input type="text" class=txt name="tel"/>
</p><p>电子邮件:
<input type="text" class=txt name="email"/>
</p><p>联系地址:
<input type="text"  class=txt name="address"/>
</p>
<p>
<input type="submit" name="submit" value="提交" class=but/>
<input type="reset" name="reset" value="清除" class=but/>
</p>
</form>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 11-20 所示，可以看到创建了一个注册表单，包含一个标题【用户注册】、【姓名】、【性别】、【年龄】、【联系方式】、【电子邮件】、【地址】等输入框和【提交】按钮等，其显示样式为默认样式。

03 添加 CSS 代码，修饰全局样式和表单样式。

```
<style>
*{
    padding:0px;
    margin:0px;
    }
body{
    font-family:"宋体";
    font-size:12px;
    }
form{
    width:300px;
    margin:0 auto 0 auto;
    font-size:12px;
    color:#999;
}
</style>
```

在 IE 11.0 中浏览效果如图 11-21 所示，可以看到页面中字体变小，其表单元素之间距离也变小。

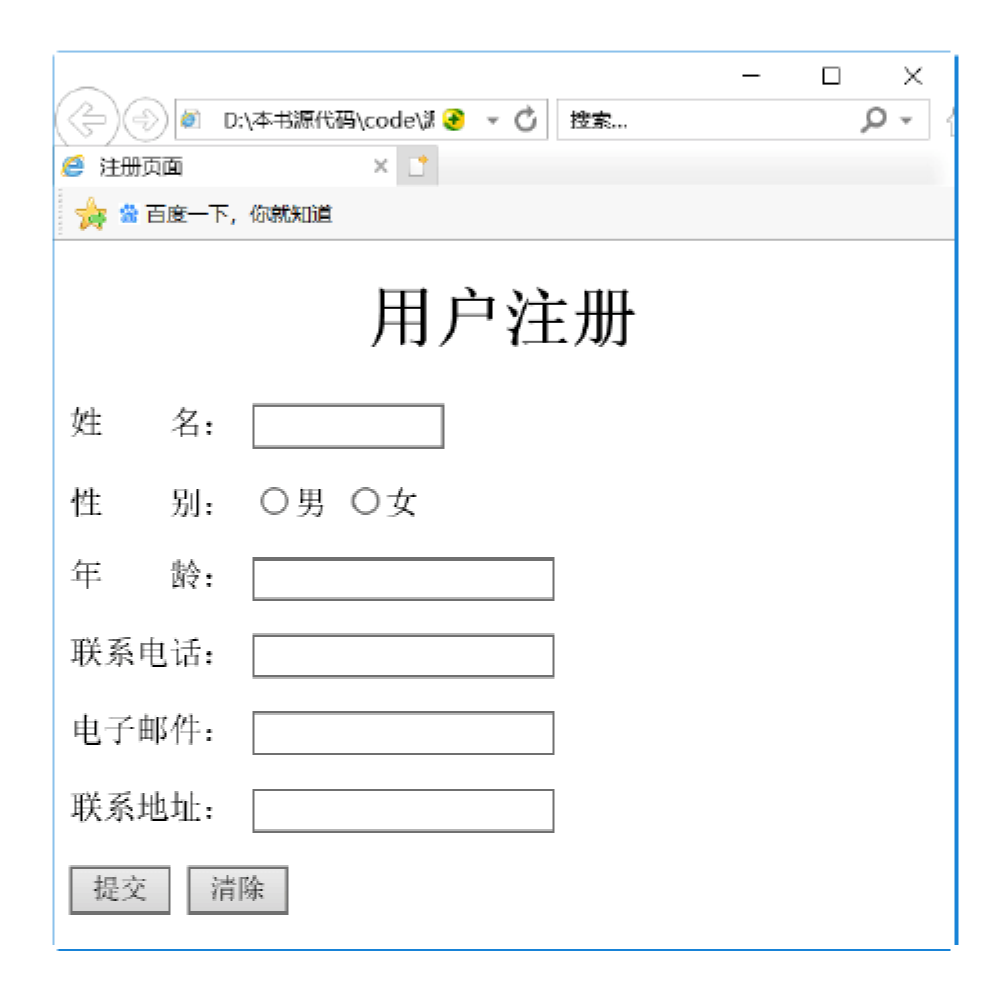


图 11-20 注册表单显示



图 11-21 CSS 修饰表单样式

04 添加 CSS 代码，修饰段落、输入框和按钮。

```
form p{
    margin:5px 0 0 5px;
       text-align:center;
    }
.txt{
    width:200px;
    background-color:#CCCCFF;
    border:#6666FF 1px solid;
    color:#0066FF;
    }
.but{
border:0px#93bee2solid;
border-bottom:#93bee21pxsolid;
border-left:#93bee21pxsolid;
border-right:#93bee21pxsolid;
border-top:#93bee21pxsolid;*/
background-color:#3399CC;
cursor:hand;
font-style:normal;
color:#cad9ea;
}
```

在 IE 11.0 中浏览效果如图 11-22 所示，可以看到表单元素有背景色，其输入字体颜色为蓝色，边框颜色为浅蓝色，按钮有边框，按钮上字体颜色为浅色。

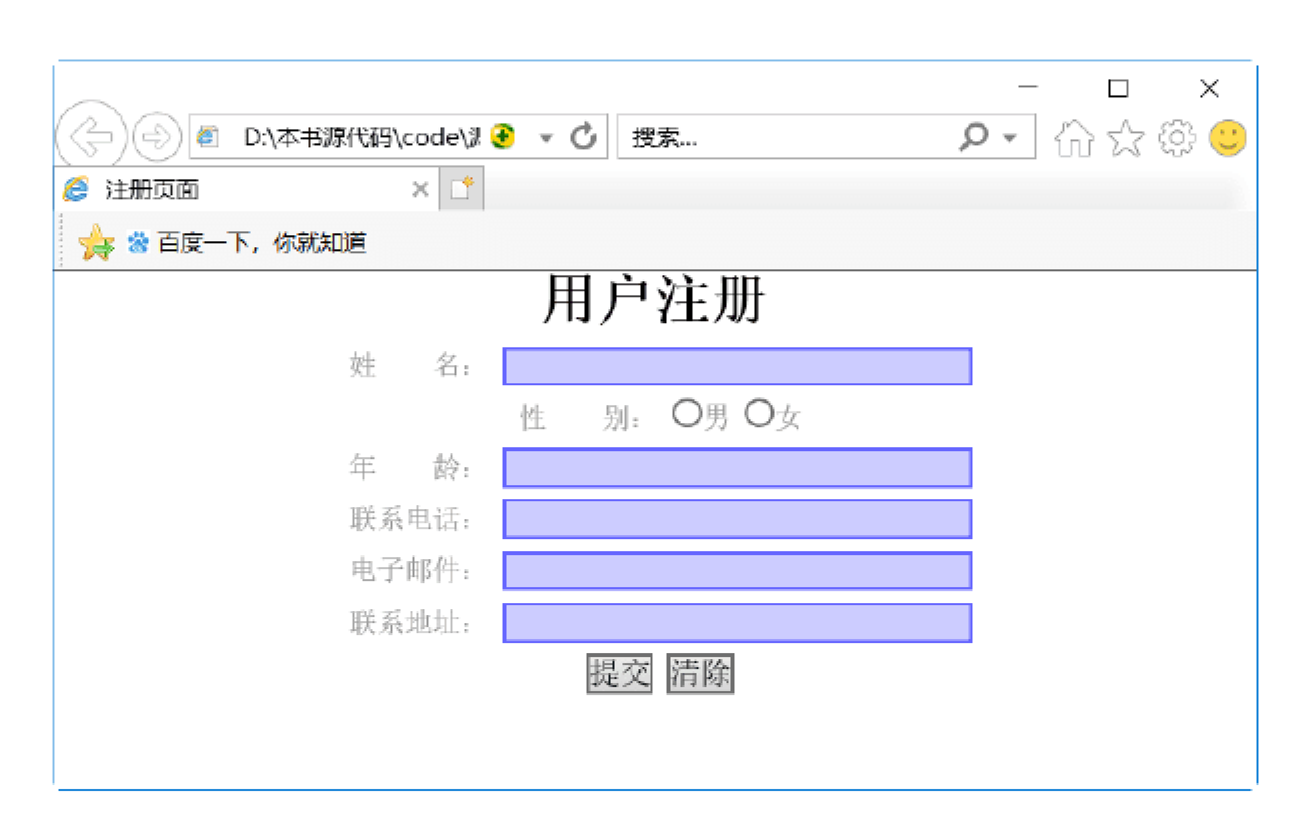


图 11-22 设置输入框和按钮样式

11.8 专家解惑

1. 构建一个表格需要注意哪些方面？

在 HTML 页面中构建表格框架时，应该尽量遵循表格的标准标记，养成良好的编写习惯，并适当利用 tab、空格和空行来提高代码的可读性，从而降低后期维护成本。特别是使用 table 表格来布局一个较大的页面时，需要在关键位置加上注释。

2. 在使用表格时会发生一些变形，这是什么原因引起的呢？

很可能是因为更改分辨率造成的，例如在 800×600 的分辨率下，一切正常，而到了 1024×800 时，则多个表格或者有的居中，有的却左排列或右排列。

表格有左、中、右三种排列方式，如果没特别进行设置，则默认为居左排列。在 800×600 的分辨率下表格恰好就有编辑区域那么宽，不容易察觉，而到了 1024×800 的时候，就发生了变形。解决的办法比较简单，即都设置为居中、居左或居右。

3. 使用 CSS 修饰表单元素时，采用默认值好还是指定好？

各个浏览器之间显示的差异，其中一个原因就是各个浏览器对部分 CSS 属性的默认值不同导致的，通常的解决办法就是指定该值，而不让浏览器使用默认值。

第12章　CSS3美化图像

一个网页如果都是文字，时间长了会让浏览者感到很枯燥，而一张恰如其分的图片，会给网页带来许多生趣。图片是直观、形象的，一张好的图片会给网页带来很高的点击率。在 CSS3 中，定义了很多属性来美化和设置图片。

12.1　图片样式

如果将多张图片直接放置到网页中，而不加以修饰，会给人一种凌乱的感觉。通过 CSS3 属性统一管理，可以定义多张图片的各种属性，如图片边框、图片缩放。

12.1.1　图片边框

在网页中放置一张图片，可以使用<img>标记，在第 2 章中已经介绍过了。当图片显示之后，其边框是否显示可以通过<img>标记中的描述属性 border 来设置。其代码如下：

```
<img src="yueji.jpg" border="3">
```

通过 HTML 标记设置图片边框，其边框显示都是黑色并且风格比较单一，唯一能够设置的就是边框的粗细，而对边框样式基本上是无能为力。这时可以采用 CSS3 对边框样式进行美化。

在 CSS3 中，使用 border-style 属性定义边框样式，即边框风格。例如可以设置边框风格为点线式边框（dotted）、破折线式边框（dashed）、直线式边框（solid）、双线式边框（double）等。

【例 12.1】（实例文件：ch12\12.1.html）

```
<!DOCTYPE html>
<html>
<head>
<title>图片边框</title>
</head>
<body>
<img src="yueji.jpg" border="3" style="border-style:dotted">
```

```
    <img src="yueji.jpg" border="3" style="border-style:dashed">
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 12-1 所示，可以看到网页显示了两张图片，其样式分别为点线式和破折线。

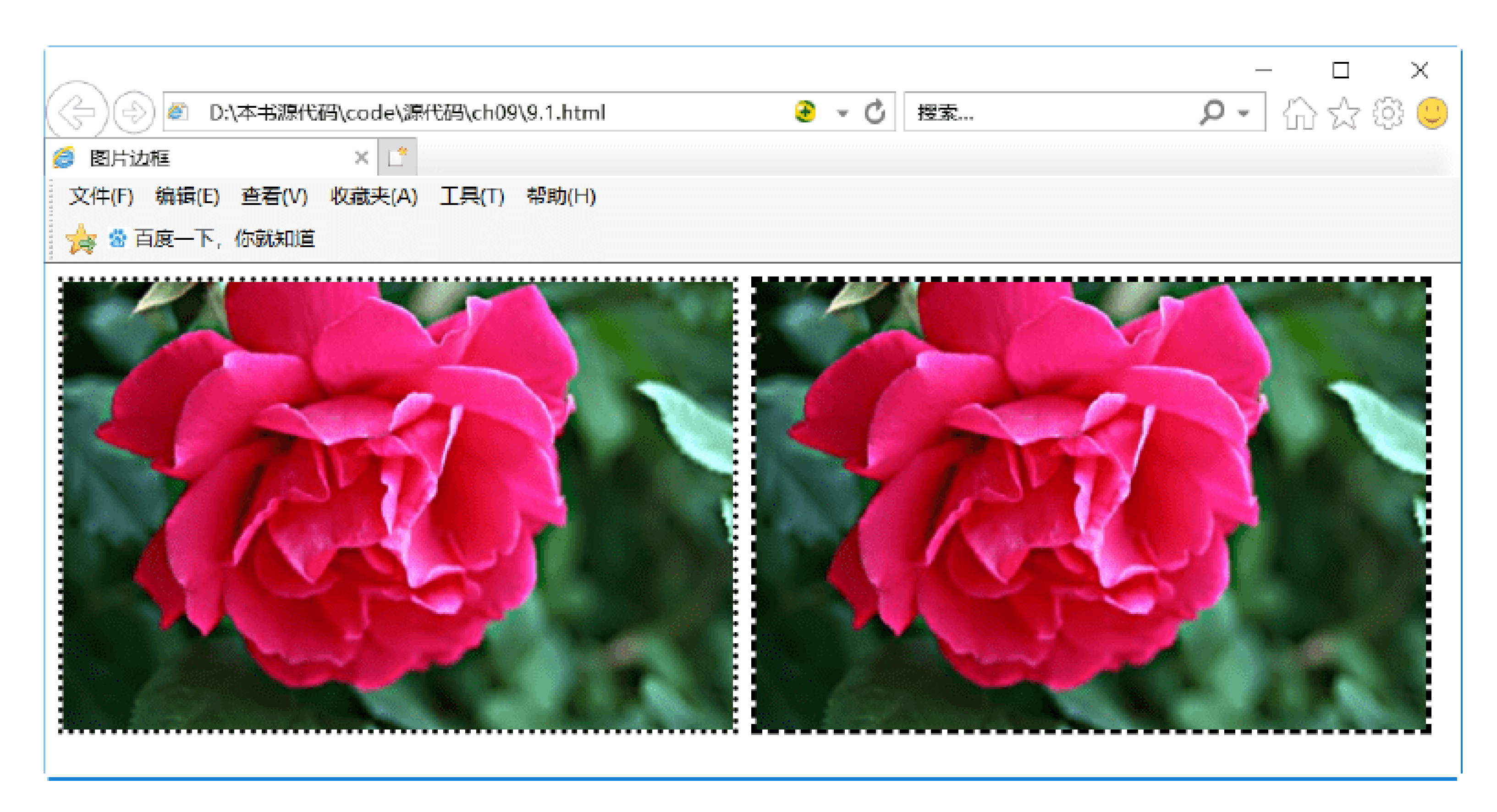


图 12-1　边框显示

另外，如果需要单独定义边框一边的样式，可以使用 border-top-style 设置上边框样式、border-right-style 设置右边框样式、border-bottom-style 设置下边框样式和 border-left-style 设置左边框样式。

【例 12.2】（实例文件：ch12\12.2.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <title>图片边框</title>
    </head>
    <body>
    <img src="yueji.jpg" border="3"
style="border-top-style:dotted;border-right-style:insert;border-bottom-styl
e:dashed;border-left-style:groove">
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 12-2 所示，可以看到网页显示了一张图片，图片的上边框、下边框、左边框和右边框分别以不同样式显示。

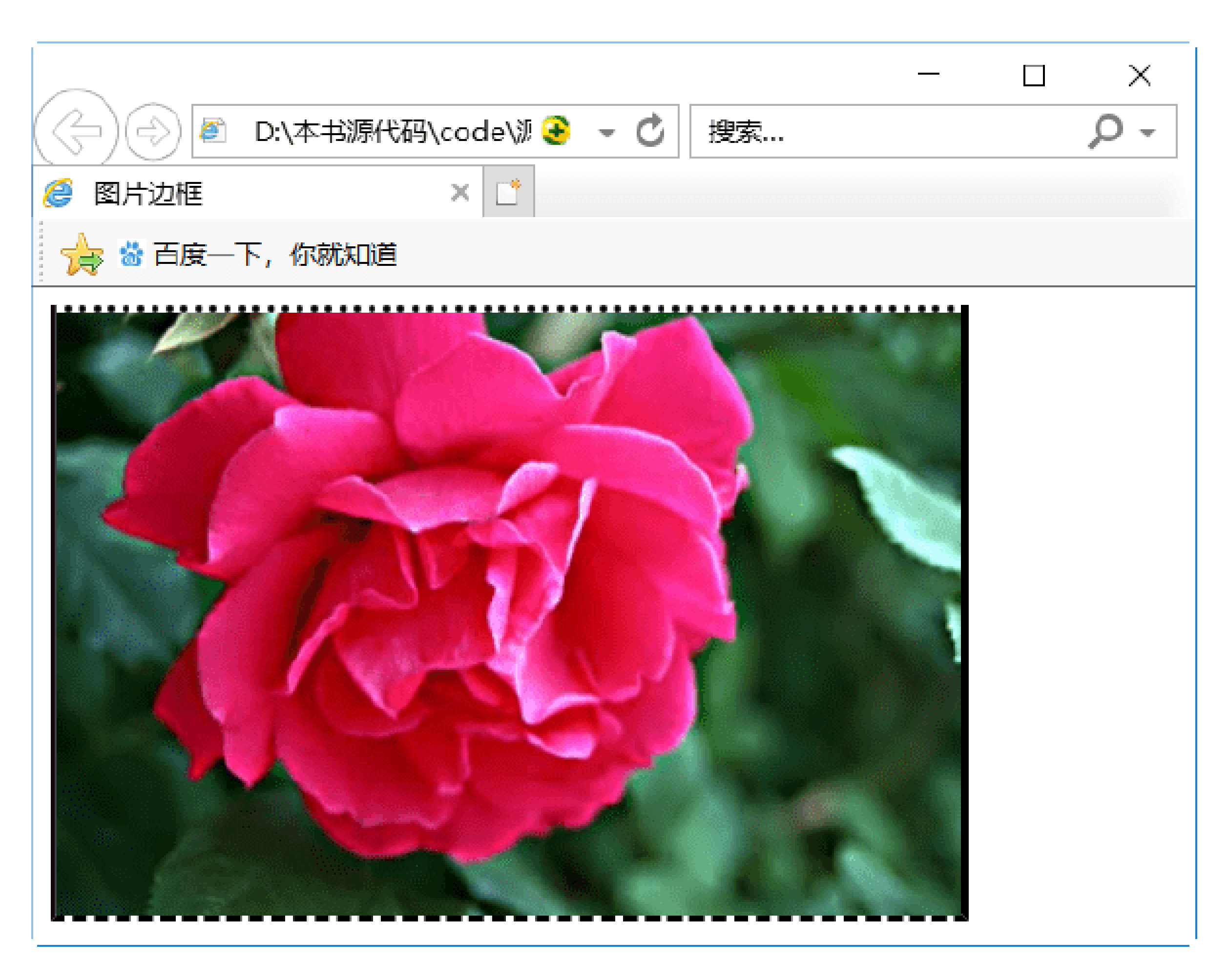


图 12-2　4 种样式边框显示

12.1.2　图片缩放

网页上显示一张图片时，默认情况下都是以图片的原始大小显示的。如果要对网页进行排版，则需要对图片的大小进行重新设置。如果对图片设置不恰当，就会造成图片的变形和失真，所以一定要保持宽度和高度属性的比例适中。对于图片大小设置，可以采用以下 3 种方式。

1. 描述标记属性 width 和 height

在 HTML 标记语言中，通过<img>标记的描述属性 height 和 width 可以设置图片大小。width 和 height 分别表示图片的宽度和高度，其中二者值可以为数值或百分比，单位是像素。需要注意的是，高度属性 height 和宽度属性 width 设置要求相同。

【例 12.3】（实例文件：ch12\12.3.html）

```
<!DOCTYPE html>
<html>
<head>
<title>图片边框</title>
</head>
<body>
<img src="feng.jpg" width=200 height=120>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-3 所示，可以看到网页显示了一张图片，其宽度为 200 像素，高度为 120 像素。

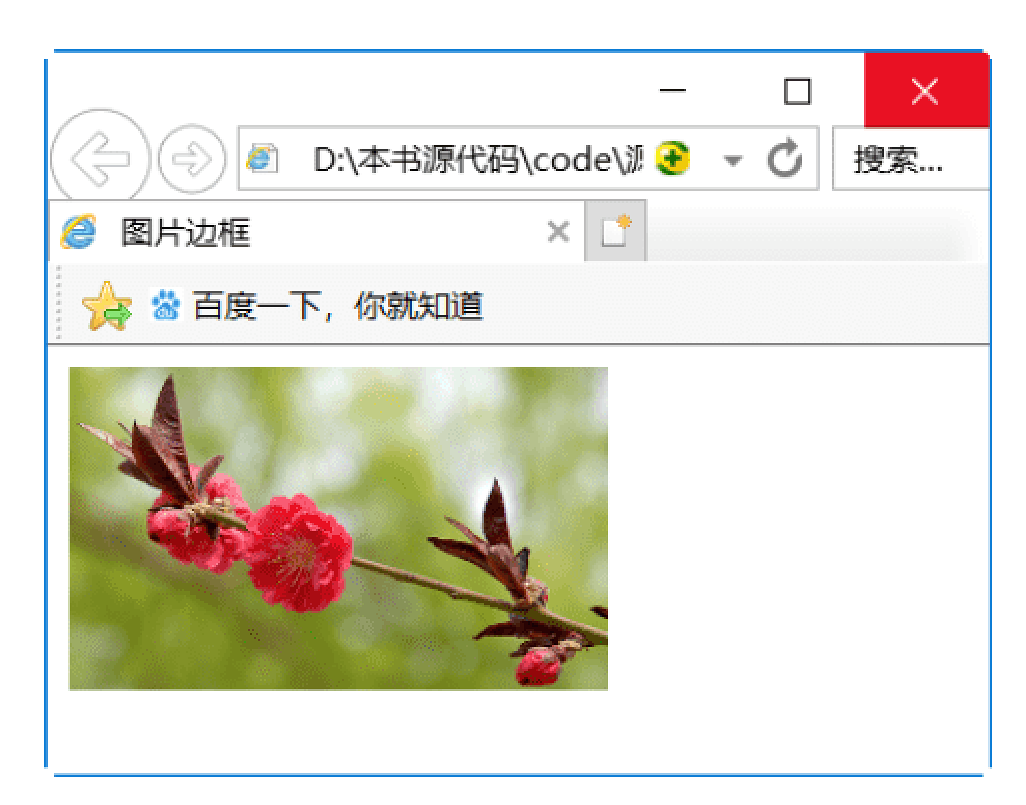


图 12-3　HTML 标记设置图片大小

2. 使用 CSS3 中 max-width 和 max-height

max-width 和 max-height 分别用来设置图片宽度最大值和高度最大值。在定义图片大小时，如果图片默认尺寸超过了定义的大小，就以 max-width 所定义的宽度值显示，而图片高度将同比例变化，定义 max-height 也是一样的道理。但是如果图片的尺寸小于最大宽度或高度，图片就按原尺寸大小显示。max-width 和 max-height 的值一般是数值类型，其语法格式如下：

```
img{
    max-height:180px;
}
```

【例 12.4】（实例文件：ch12\12.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>图片边框</title>
<style>
img{
    max-height:180px;
}
</style>
</head>
<body>
<img src="feng.jpg">
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-4 所示，可以看到网页显示了一张图片，其显示高度是 180 像素，宽度将作同比例缩放。

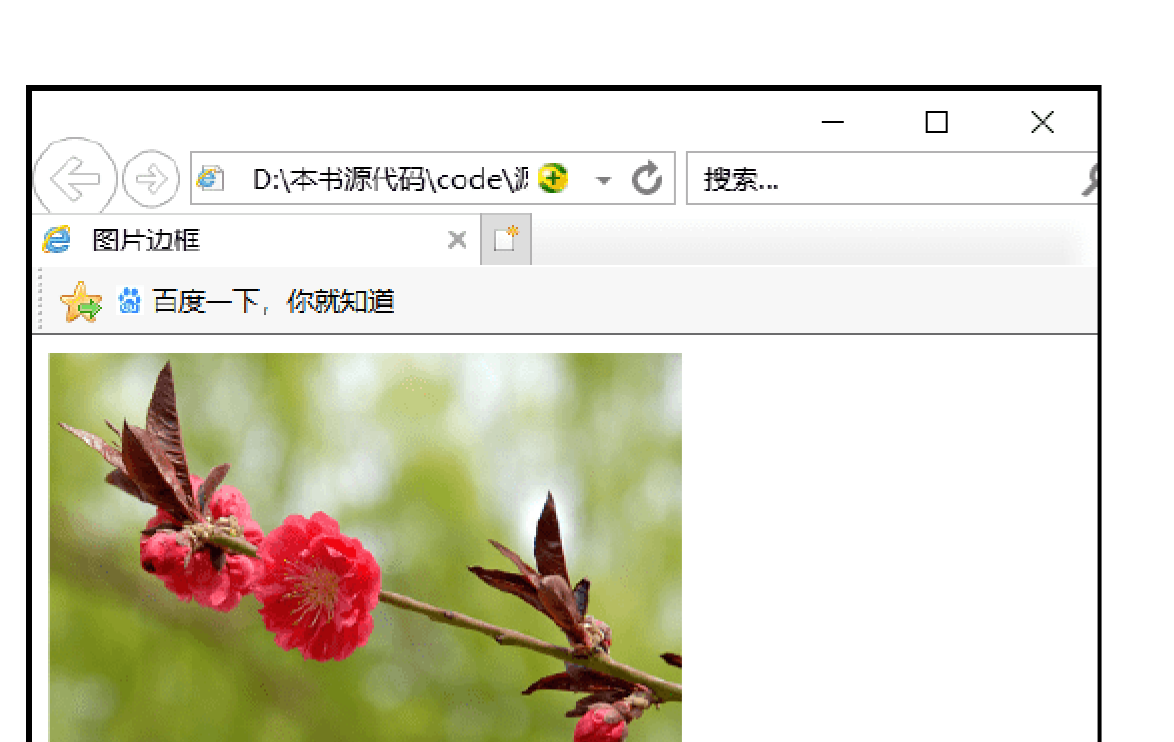


图 12-4　同比例缩放图片

在本例中，也可以只设置 max-width 来定义图片最大宽度，而让高度自动缩放。

3. 使用 CSS3 中 width 和 height

在 CSS3 中，可以使用属性 width 和 height 来设置图片宽度和高度，从而达到对图片的缩放效果。

【例 12.5】（实例文件：ch12\12.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>图片边框</title>
</head>
<body>
<img src="feng.jpg">
<img src="feng.jpg" style="width:90px;height:100px">
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-5 所示，可以看到网页显示了两张图片，第一张图片以原始大小显示，第二张图片以指定大小显示。

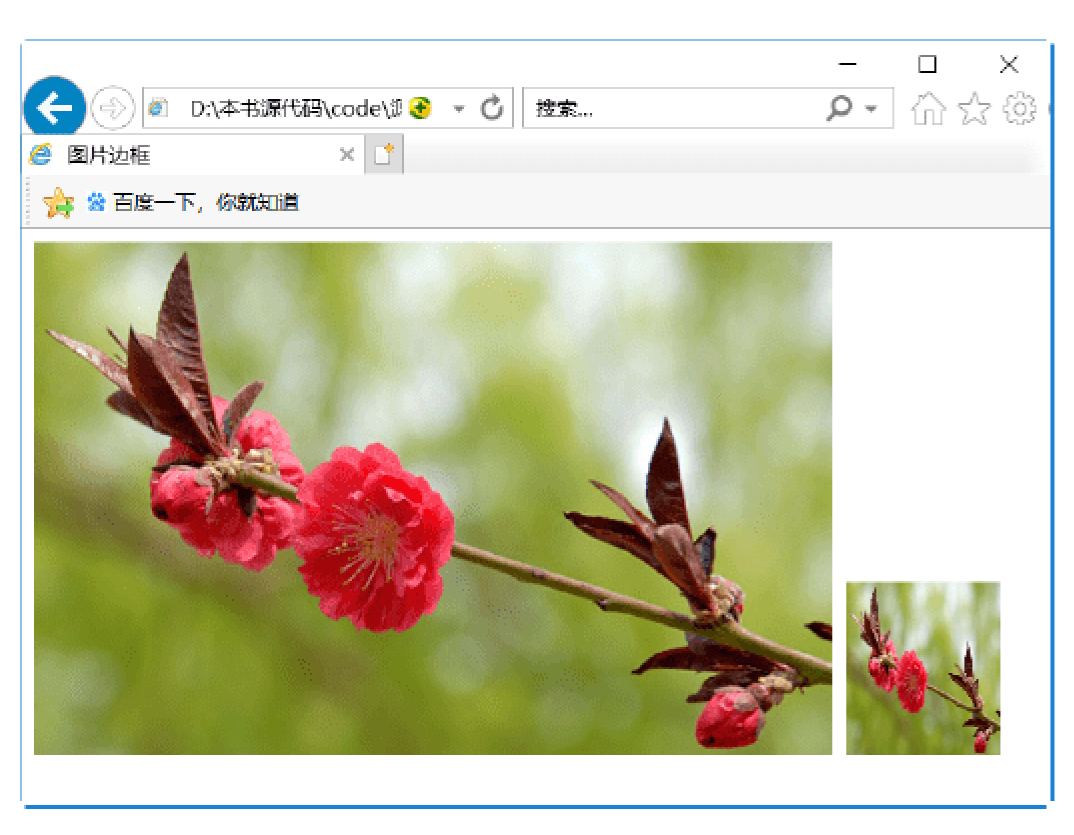


图 12-5　CSS 指定图片大小



需要注意的是，当仅仅设置了图片的 width 属性，而没有设置 height 属性时，图片本身会自动等纵横比例缩放，如果只设置 height 属性也是一样的道理。只有当同时设置 width 和 height 属性时才会不等比例缩放。

12.2　对齐图片

一个凌乱的图文网页是每一个浏览者都不喜欢看的，而一个图文并茂、排版格式整洁简约的页面更容易让网页浏览者接受，可见图片的对齐方式是非常重要的。本节将介绍使用 CSS3 属性定义图文对齐方式。

12.2.1　横向对齐方式

所谓图片横向对齐，就是在水平方向上进行对齐，其对齐样式和文字对齐比较相似，都有 3 种对齐方式，分别为左对齐、居中对齐和右对齐。

如果要定义图片对齐方式，则不能在样式表中直接定义图片样式，而是需要在图片的上一个标记级别（父标记）定义对齐方式，让图片继承父标记的对齐方式。之所以这样定义父标记对齐方式，是因为 img（图片）本身没有对齐属性，需要使用 CSS 继承父标记的 text-align 来定义对齐方式。

【例 12.6】（实例文件：ch12\12.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>图片横向对齐</title>
</head>
<body>
<p style="text-align:left"><img src="mudan.jpg" style="max-width:140px;">
图片左对齐</p>
<p style="text-align:center"><img src="mudan.jpg"
style="max-width:140px;">图片居中对齐</p>
<p style="text-align:right"><img src="mudan.jpg"
style="max-width:140px;">图片右对齐</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-6 所示，可以看到网页显示了三张图片，大小相同，但对齐方式分别是左对齐、居中对齐和右对齐。

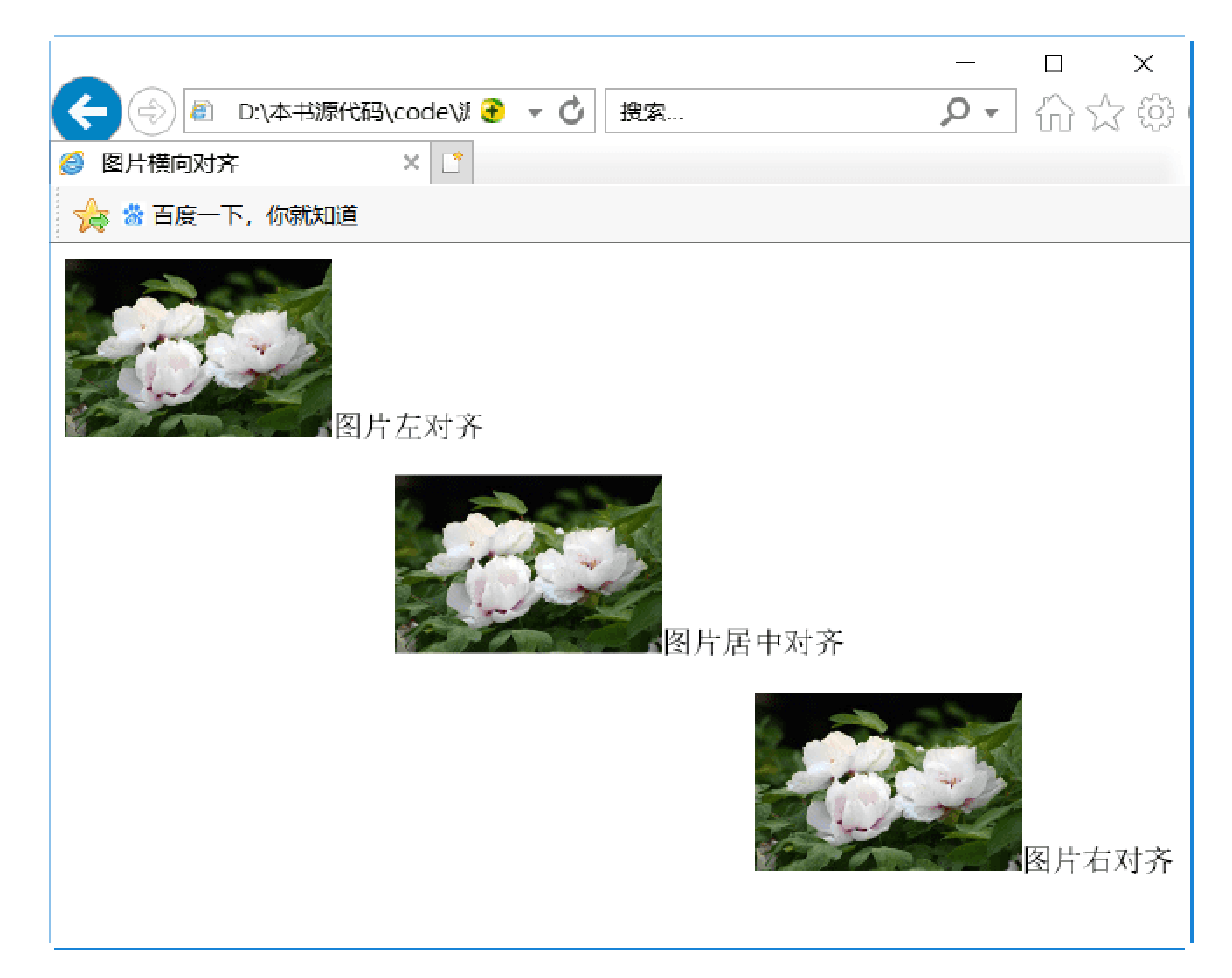


图 12-6　图片横向对齐

12.2.2　纵向对齐方式

纵向对齐就是垂直对齐，即在垂直方向上和文字进行搭配使用。通过对图片垂直方向上的设置，可以设置图片和文字的高度一致。在 CSS3 中，对于图片纵向设置，通常使 vertical-align 属性来定义。

vertical-align 属性设置元素的垂直对齐方式，即定义行内元素的基线相对于该元素所在行的基线的垂直对齐。允许指定负的长度值和百分比值，这会使元素降低。在表单元格中，这个属性会设置单元格框中内容的对齐方式，其语法格式如下：

```
    vertical-align:baseline|sub|super|top|text-top|middle|bottom|text-bottom|length
```

上面参数含义如表 12-1 所示。

表 12-1　纵向对齐参数含义

参数名称	说明
baseline	支持 valign 特性的对象内容与基线对齐
sub	垂直对齐文本的下标
super	垂直对齐文本的上标
top	将支持 valign 特性的对象的内容与对象顶端对齐
text-top	将支持 valign 特性的对象的文本与对象顶端对齐
middle	将支持 valign 特性的对象的内容与对象中部对齐

（续表）

参数名称	说明
bottom	将支持 valign 特性的对象的文本与对象底端对齐
text-bottom	将支持 valign 特性的对象的文本与对象底端对齐
length	由浮点数字和单位标识符组成的长度值或百分数，可为负数，定义由基线算起的偏移量。基线对于数值来说为 0，对于百分数来说就是 0

【例 12.7】（实例文件：ch12\12.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title>图片纵向对齐</title>
<style>
img{
max-width:100px;
}
</style>
</head>
<body>
<p>纵向对齐方式:baseline<img src=mudan.jpg
style="vertical-align:baseline"></p>
<p>纵向对齐方式:bottom<img src=mudan.jpg
style="vertical-align:bottom"></p>
<p>纵向对齐方式:middle<img src=mudan.jpg
style="vertical-align:middle"></p>
<p>纵向对齐方式:sub<img src=mudan.jpg style="vertical-align:sub"></p>
<p>纵向对齐方式:super<img src=mudan.jpg style="vertical-align:super"></p>
<p>纵向对齐方式:数值定义<img src=mudan.jpg style="vertical-align:20px"></p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-7 所示，可以看到网页显示 6 张图片，垂直方向上分别是 baseline、bottom、middle、sub、super 和数值对齐。



仔细观察图片和文字的不同对齐方式，即可深刻理解各种纵向对齐的不同之处。

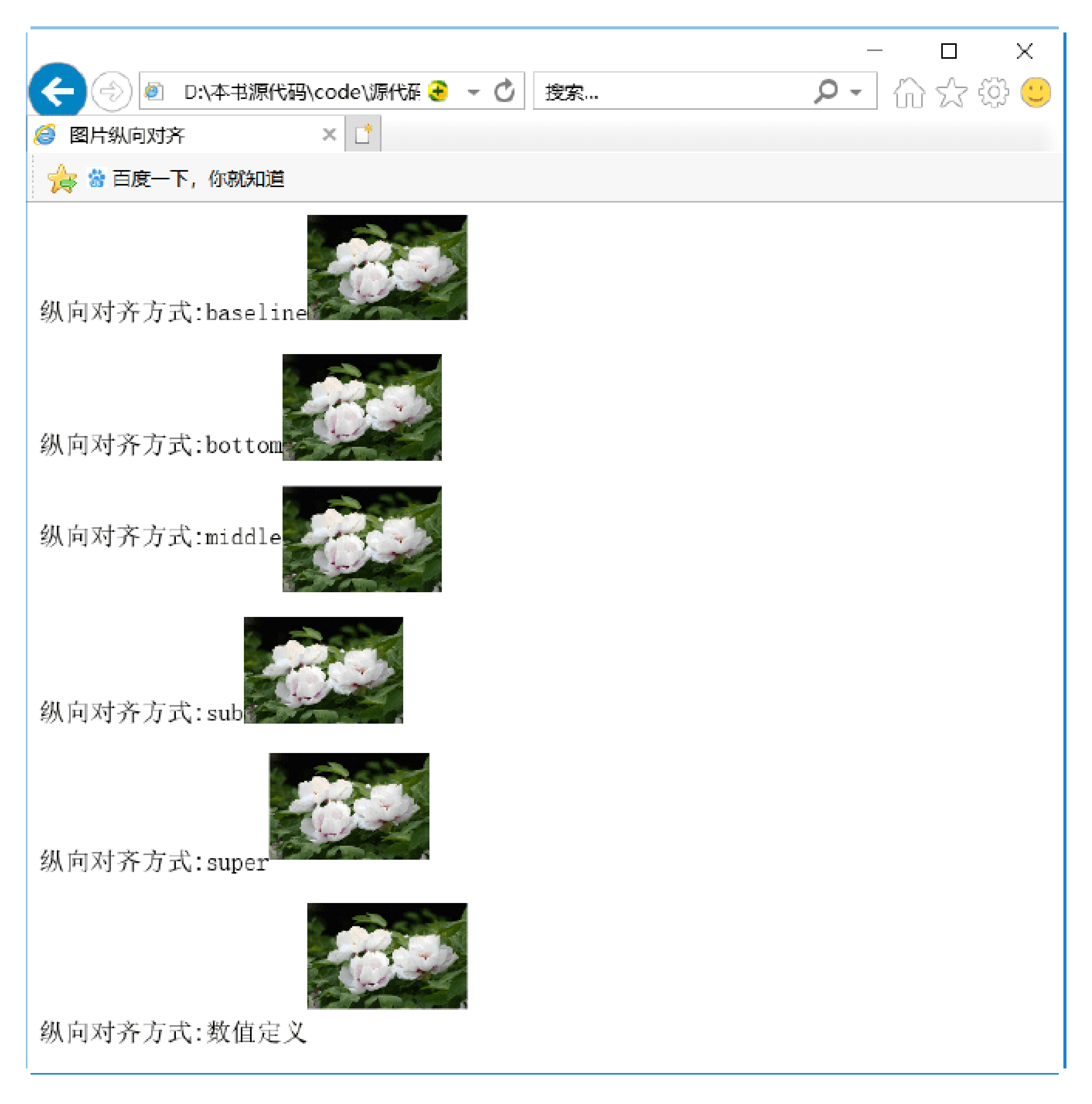


图 12-7　图片纵向对齐

12.3　图文混排

排版一个网页，比较常见的方式就是图文混排。文字说明主题，图像显示新闻情境，二者结合起来相得益彰。本节将介绍图片和文字的排版方式。

12.3.1　文字环绕

在网页中进行排版时，可以将文字设置成环绕图片的形式，即文字环绕。文字环绕应用非常广泛，如果再配合背景，就可以达到绚丽的效果。

在 CSS3 中，可以使用 float 属性来定义该效果。float 属性主要定义元素在哪个方向浮动。一般情况下这个属性应用于图像元素，使文本围绕在图像周围，有时它也可以定义其他元素浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。如果浮动非替换元素，则要指定一个明确的宽度，否则它们会尽可能地窄。

float 语法格式如下：

```
float:none|left|right
```



其中 none 表示默认值对象不漂浮；left 表示文本流向对象的右边；right 表示文本流向对象的左边。

【例 12.8】（实例文件：ch12\12.8.html）

```
<!DOCTYPE html>
<html>
<head>
<title>文字环绕</title>
<style>
img{
max-width:120px;
float:left;
}
</style>
</head>
<body>
<p>
一个可爱的小家伙乘着风儿顽皮地落在了我的肩上，我低头看了看，原来是一片枫叶。
<img src="fengye.jpg">
我小心翼翼地把它捧在手中，一阵风儿吹过，叶子的几个小尖角随风摆起，多像婴儿的小手掌啊！平滑的叶面，清晰的脉络，十分柔软细嫩。枫叶树种在秋冬的时候，体内会产生一些化学反应，让原本树叶中所含枫叶(10片)的物质或部分组织分解之后，回收储藏在茎或根的部位，来年春天的时候可以再利用。叶绿体、叶绿素就是被分解回收的对象之一，因为叶绿素的含量较大而遮盖了其他颜色，使叶片呈绿色，因此当叶子里的叶绿素没有的时候，其他色素的颜色就彰显出来，如花青素的红色、胡萝卜素的黄色和叶黄素的黄色等。除此之外，枫叶中贮存的糖分还会分解转变成花青素，使叶片的颜色更加艳丽、火红。 枫叶没有五个“手指”就不是枫叶，而且枫叶的“五指”上具有锯齿，这是枫叶的特色！
</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-8 所示，可以看到图片被文字所环绕，并在文字的左方向显示。如果将 float 属性的值设置为 right，则其图片会在文字右方显示并环绕。

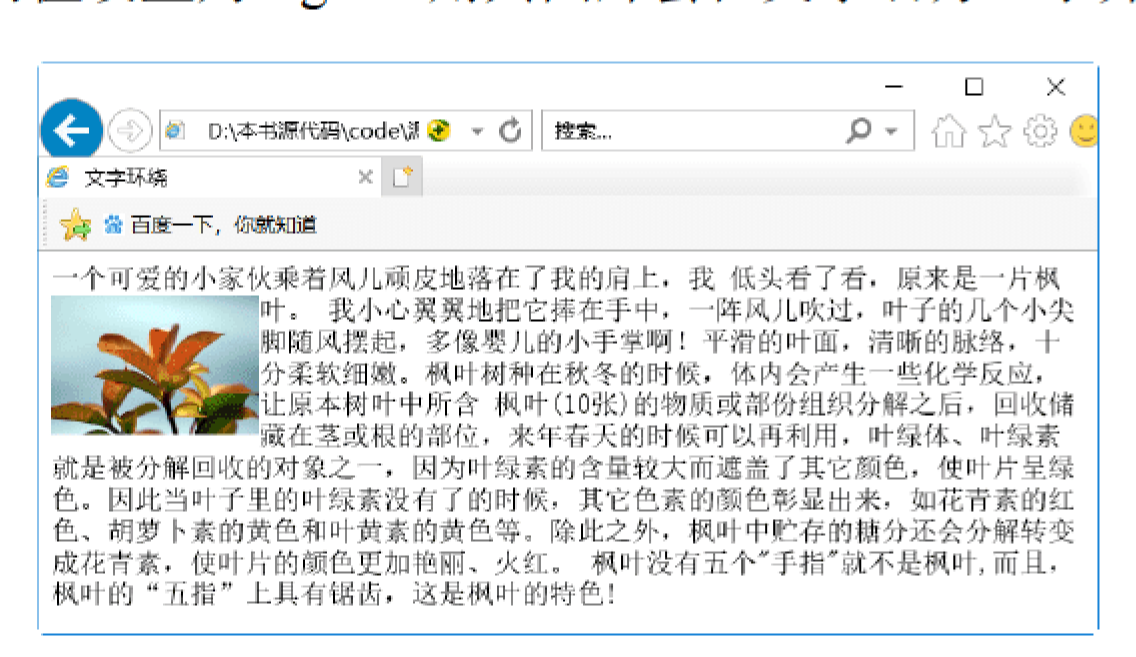


图 12-8　文字环绕效果

12.3.2　设置图片与文字间距

如果需要设置图片和文字之间的距离（即文字之间存在一定间距，而不是紧紧环绕），就可以使用 CSS3 中的属性 padding 来设置。

padding 属性主要用来在一个声明中设置所有内边距属性，即可以设置元素所有内边距的

宽度，或者设置各边上内边距的宽度。如果一个元素既有内边距又有背景，则从视觉上看可能会延伸到其他行，有可能还会与其他内容重叠。设置时不允许指定负边距值。

其语法格式如下：

```
padding:padding-top|padding-right|padding-bottom|padding-left
```

其参数 padding-top 用来设置距离顶部内边距；padding-right 用来设置距离右部内边距；padding-bottom 用来设置距离底部内边距；padding-left 用来设置距离左部内边距。

【例 12.9】（实例文件：ch12\12.9.html）

```
<!DOCTYPE html>
<html>
<head>
<title>文字环绕</title>
<style>
img{
max-width:120px;        /*设置图像的最大宽度*/
float:left;             /*设置图像靠左显示*/
padding-top:10px;       /*设置图像的上内边距*/
padding-right:50px;   /*设置图像的右内边距*/
padding-bottom:10px;  /*设置图像的下内边距*/
}
</style>
</head>
<body>
<p>一个可爱的小家伙乘着风儿顽皮地落在了我的肩上，我低头看了看，原来是一片枫叶。
<img src="fengye.jpg">
我小心翼翼地把它捧在手中，一阵风儿吹过，叶子的几个小尖角随风摆起，多像婴儿的小手掌啊！平滑的叶面，清晰的脉络，十分柔软细嫩。枫叶树种在秋冬的时候，体内会产生一些化学反应，让原本树叶中所含枫叶(10片)的物质或部分组织分解之后，回收储藏在茎或根的部位，来年春天的时候可以再利用。叶绿体、叶绿素就是被分解回收的对象之一，因为叶绿素的含量较大而遮盖了其他颜色，使叶片呈绿色，因此当叶子里的叶绿素没有的时候，其他色素的颜色就彰显出来，如花青素的红色、胡萝卜素的黄色和叶黄素的黄色等。除此之外，枫叶中贮存的糖分还会分解转变成花青素，使叶片的颜色更加艳丽、火红。 枫叶没有五个“手指”就不是枫叶，而且枫叶的“五指”上具有锯齿，这是枫叶的特色!</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-9 所示，可以看到图片被文字所环绕，并且文字与图片右边间距为 50 像素，上下各为 10 像素。

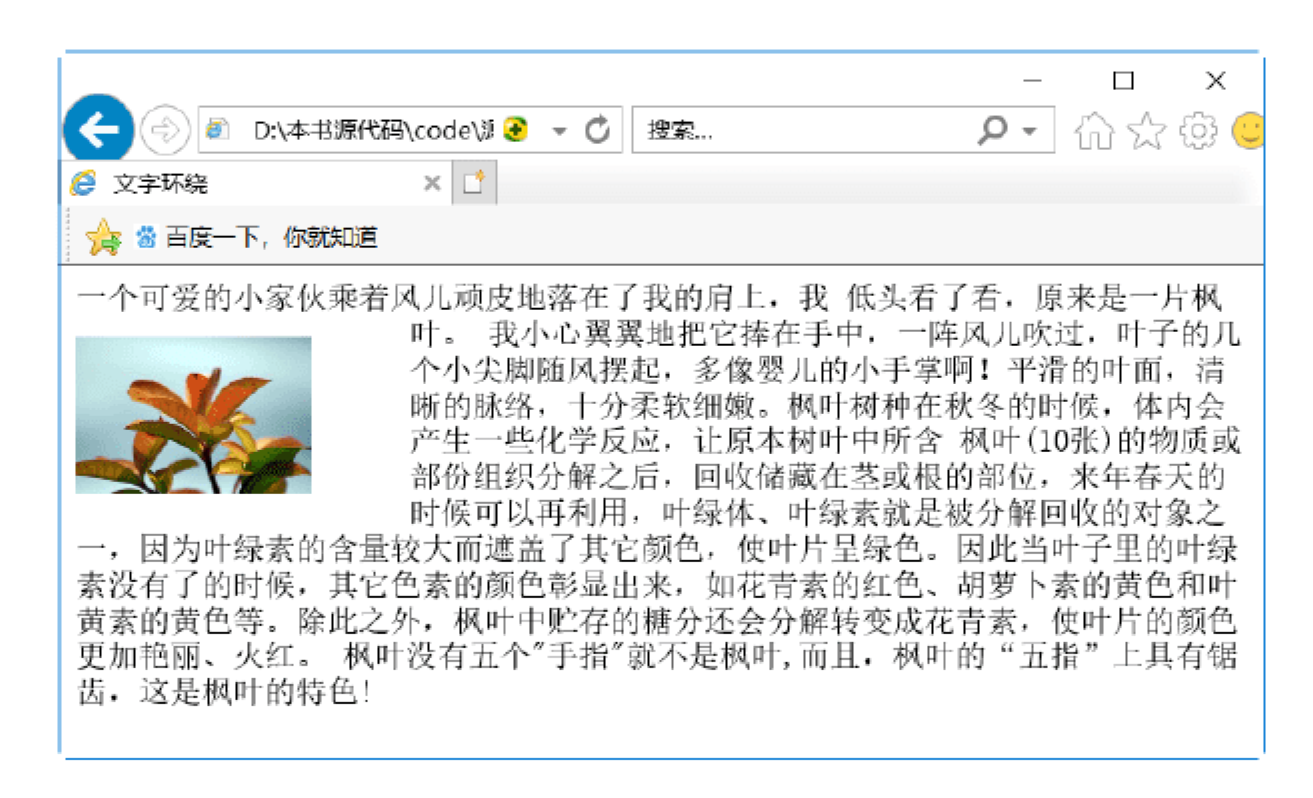


图 12-9　设置图片和文字边距

12.4　综合实例 1——一句话新闻

在各大网站中，点击率最高的通常是新闻，因为人们每天都在不停地浏览新闻。新闻格式要求简洁明了、文字表达清楚，配上图片更是图文并茂。对于网页新闻排版，应根据其新闻内容，可以有一句话新闻。本实例将介绍如何配合图片，设计出一句话新闻的网页版面，具体步骤如下：

01 分析需求。

在本实例中，如果要显示一句话新闻，则需要包含两个部分：一个是新闻标题；另一个是新闻内容，新闻内容可以是图片和段落文字。此处可以使用 div 将两个部分分成不同的层次。

02 构建 HTML 页面。

页面中包含的这两部分可以使用 3 个<div>标记来进行层次划分。一个 div 包含整个一句话新闻；一个 div 包含标题（标题可以分为正标题和副标题）；一个 div 包含图片和段落。其代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>时事新闻</title>
<style>
</style>
</head>
<body>
<div>
<div>
<p>英国皇家国际航展开幕</p>
```

```
<p>2011-07-17 09:38 来源：新华网</p>
</div>
<div>
<p align=center>
<img src=feiji.jpg border=1>
<p>
<p>
7月16日，在英国的费尔福德，一架A-10攻击机进行飞行表演。为期2天的英国皇家国际航展当日
在费尔福德空军基地开幕，这是世界上规模最大的军用飞机航空展之一。
</p>
</div>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-10 所示，可以看到在网页中显示了段落、图片和标题。

03 添加 CSS 代码，修饰整体 div。

```
<style>
.da{border:#0033FF 1px solid;}
</style>
```

04 在 HTML 代码中，使用类标识符指向 da。

```
<div class=da>
<div >
<p >英国皇家国际航展开幕</p>
<p >2011-07-17 09:38 来源：新华网</p>
</div>
<div>
<p align=center>
<img src=feiji.jpg border=1>
<p>
7月16日，在英国的费尔福德，一架A-10攻击机进行飞行表演。为期2天的英国皇家国际航展当日
在费尔福德空军基地开幕，这是世界上规模最大的军用飞机航空展之一。
</p>
</div>
</div>
```

05 在 IE 11.0 中浏览效果如图 12-11 所示，可以看到在网页中显示了一个边框，并且段落、图片包含在边框里面。



图 12-10　基本层次划分



图 12-11　CSS 整体修饰

06 添加 CSS 代码，修饰正标题和副标题。

```
.title{color:blue;font-size:25px;text-align:center}  /*设置正标题样式*/
.xtitle{ /*设置副标题样式*/
     text-align:center;
     font-size:13px;
     color:gray;
}
```

07 在 HTML 代码中，引用上面两个类标识符，代码如下：

```
<div class=da>
<div>
<p class=title>英国皇家国际航展开幕</p>
<p class=xtitle>2011-07-17 09:38 来源：新华网</p>
</div>
<div>
<p align=center>
<img src=feiji.jpg border=1>
<p>
<p>
7月16日，在英国的费尔福德，一架A-10攻击机进行飞行表演。为期2天的英国皇家国际航展当日在费尔福德空军基地开幕，这是世界上规模最大的军用飞机航空展之一。
</p>
</div>
</div>
```

08 在 IE 11.0 中浏览效果如图 12-12 所示，可以看到在网页中正标题和副标题都居中显示，并且正标题以蓝色显示，大小为 25 像素；副标题以灰色显示，大小为 13 像素。

09 添加 CSS 代码，修饰图片，代码如下：

```
img{
    border-top-style:solid;     /*设置图像的上边框样式为实线*/
    border-right-style:dashed;  /*设置图像的右边框样式为虚线*/
```

```
    border-bottom-style:solid; /*设置图像的下边框样式为实线*/
    border-left-style:dashed; /*设置图像的左边框样式为虚线*/
}
```

10 此处使用了标记标识符，会直接作用于网页中的图片，此处就不再显示 HTML 代码了。在 IE 11.0 中浏览效果如图 12-13 所示，可以看到在网页中图片边框显示了不同样式，上下以直线显示，左右以破折线显示。



图 12-12　修饰标题



图 12-13　图片边框样式修饰

11 添加 CSS 代码，修饰段落，代码如下：

```
    <p style="text-indent:10mm;font-size:15px;"> 7月16日，在英国的费尔福德，一架A-10攻击机进行飞行表演。为期2天的英国皇家国际航展当日在费尔福德空军基地开幕，这是世界上规模最大的军用飞机航空展之一。
    </p>
```

12 在 IE 11.0 中浏览效果如图 12-14 所示，可以看到在网页中段落缩进 10mm，并且字体大小为 15 像素。



图 12-14　修饰段落

12.5 综合实例 2——学校宣传单

每年暑假时，学校招生就铺天盖地，大量的宣传单页到处都是。本实例模仿一个学校宣传单，进行图文混排，从而加深前面学习的知识，具体步骤如下：

01 分析需求。

本实例包含两个部分：一个部分是图片信息，用来介绍学校场景；另一个部分是段落信息，用来介绍学校历史和理念。这两部分都放在一个 div 中。

02 构建 HTML 网页。

创建 HTML 页面，页面中包含一个 div，div 中又包含图片和两个段落信息。其代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>学校宣传单</title>
</head>
<body>
<div>
<img src="xuexiao.jpg"/><p>某大学风景优美</p><p> 学校发扬“百折不挠、艰苦创业”的办学传统，坚持“质量立校、人才兴校、创新强校、文化铸校、和谐荣校”的办学理念，弘扬“爱国荣校、民主和谐、求真务实、开放创新”的精神</p>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 12-15 所示，可以看到在网页中标题和内容被一条虚线隔开。

03 添加 CSS 代码，修饰 div。

```
<style>
big{
width:430px;
}
</style>
```

在 HTML 代码中，将 big 引用到 div 中，代码如下：

```
<div class=big>
    <img src="xuexiao.jpg"/><p>某大学风景优美</p><p> 学校发扬“百折不挠、艰苦创业”的办学传统，坚持“质量立校、人才兴校、创新强校、文化铸校、和谐荣校”的办学理念，弘扬“爱国荣校、民主和谐、求真务实、开放创新”的精神</p>
</div>
```

在 IE 11.0 中浏览效果如图 12-16 所示，可以看到在网页中段落以块的形式显示。



图 12-15　HTML 页面显示



图 12-16　修饰 div 层

04 添加 CSS 代码，修饰图片。

```
img{
  width:260px;                  /*设置图像的宽度*/
  height:220px;                 /*设置图像的高度*/
  border:#009900 2px solid;     /*设置图像的边框颜色、粗细和实线*/
  float:left;                   /*设置图像向左浮动*/
  padding-right:0.5px;          /*设置右内边距的大小*/
}
```

在 IE 11.0 中浏览效果如图 12-17 所示，可以看到在网页中图片以指定大小显示，并且带有边框，图片在左侧被文字环绕。

05 添加 CSS 代码，修饰段落。

```
p{
  font-family:"宋体";   /*设置段落文字的字体*/
  font-size:14px;       /*设置段落文字的大小*/
  line-height:20px;     /*设置段落文字的行高*/
}
```

在 IE 11.0 中浏览效果如图 12-18 所示，可以看到在网页中段落以宋体显示，大小为 14 像素，行高为 20 像素。



图 12-17　修饰图片



图 12-18　修饰段落

12.6　专家解惑

1. 网页进行图文排版时，哪些是必须要做的？

在进行图文排版时，通常有如下 5 个方面需要网页设计者考虑。

（1）首行缩进：段落的开头应该空两格。HTML 中空格键起不了作用，当然可以用“nbsp;”来代替一个空格，但这不是理想的方式。应该用 CSS3 中的首行缩进，且大小为 2cm。

（2）图文混排：在 CSS3 中，可以用 float 来让文字在没有清理浮动的时候显示在图片以外的空白处。

（3）设置背景色：设置网页背景以增加效果。

（4）文字居中：可以 CSS 的 text-align 设置文字居中。

（5）显示边框：通过 border 为图片添加一个边框。

2. 设置文字环绕时，float 元素为什么会失去作用？

很多浏览器在显示未指定 width 的 float 元素时会发生错误。所以不管 float 元素的内容如何，一定要为其指定 width 属性。

第13章　CSS3美化背景与边框

任何一个页面，首先映入眼帘的就是网页的背景色和基调，不同类型的网站有不同的背景和基调，因此页面中的背景通常是网站设计时一个重要的步骤。对于单个 HTML 元素来说，可以通过 CSS3 属性设置元素边框样式，包括宽度、显示风格和颜色等。本章将重点介绍网页背景设置和 HTML 元素边框样式。

13.1　背景相关属性

背景是网页设计中的重要因素之一，一个背景优美的网页总能吸引不少访问者。例如，喜庆类网站都是以火红背景为主题，CSS 的强大表现功能在背景方面同样发挥得淋漓尽致。

13.1.1　背景颜色

background-color 属性用于设置网页背景色，同设置前景色的 color 属性一样，background-color 属性接受任何有效的颜色值，而对于没有设置背景色的标记，默认背景色为透明（tranaparent）。

其语法格式如下：

```
{background-color:transparent|color}
```

关键字 transparent 是个默认值，表示透明。背景颜色 color 的设置方法可以采用英文单词、十六进制、RGB、HSL、HSLA 和 GRBA。

【例 13.1】（实例文件：ch13\13.1.html）

```
<!DOCTYPE html>
<html>
<head><title>背景色设置</title></head>
<body style="background-color:PaleGreen;color:Blue">
<p>background-color属性设置背景色，color属性设置字体颜色，即前景色。</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-1 所示，可以看到网页背景色显示浅绿色，而字体颜色为蓝色。注意，在网页设计时其背景色不要使用太艳的颜色，否则会给人一种喧宾夺主的感觉。

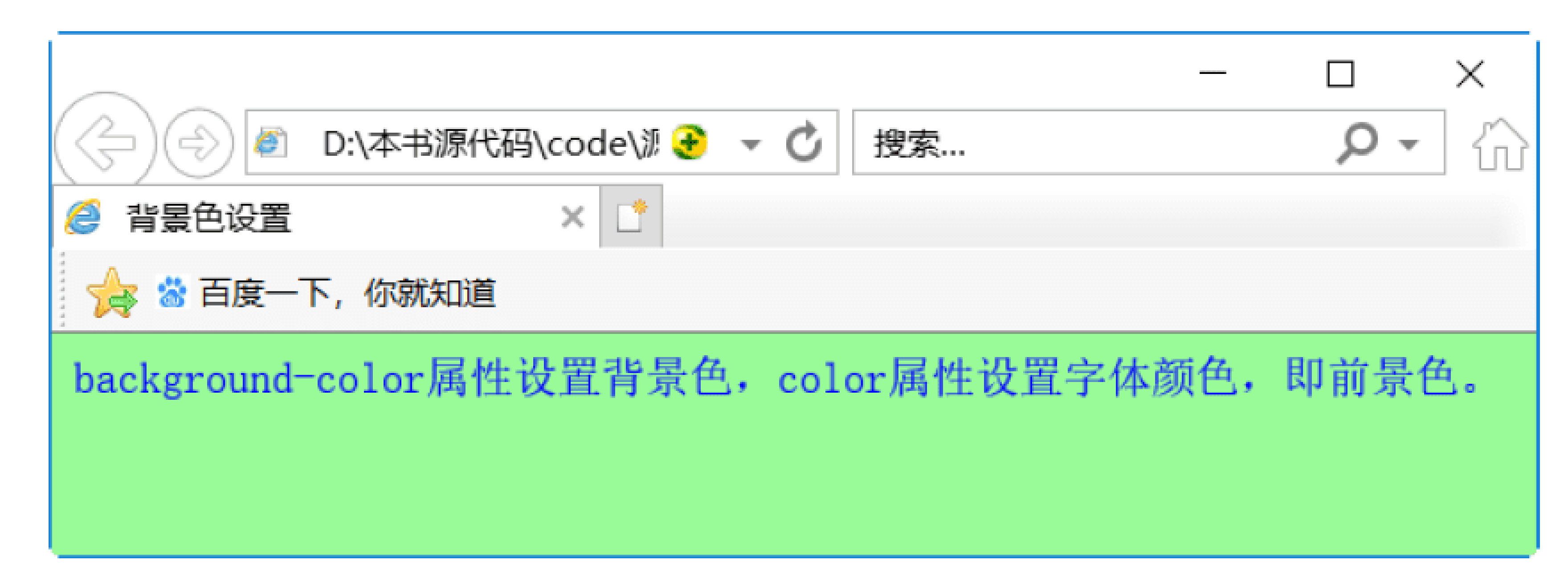


图 13-1　设置背景色

background-color 不仅可以设置整个网页的背景颜色，而且还可以设置指定 HTML 元素的背景色，如设置 h1 标题的背景色、段落 p 的背景色等。

【例 13.2】（实例文件：ch13\13.2.html）

```
<!DOCTYPE html>
<html>
<head>
<title>背景色设置</title>
<style>
h1 {
    background-color:red;  /*设置标题的背景颜色为红色*/
    color:black;           /*设置标题的颜色为黑色*/
   text-align:center;      /*设置标题的居中显示*/
}
p{
     background-color:gray;  /*设置正文的背景颜色为灰色*/
     color:blue;             /*设置正文的颜色为蓝色*/
     text-indent:2em;        /*设置文本缩进*/
}
</style>
</head>
<body>
<h1>颜色设置</h1>
<p>background-color属性设置背景色，color属性设置字体颜色，即前景色。</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-2 所示，可以看到网页中标题区域背景色为红色，段落区域背景色为灰色，并且分别为字体设置了不同的前景色。

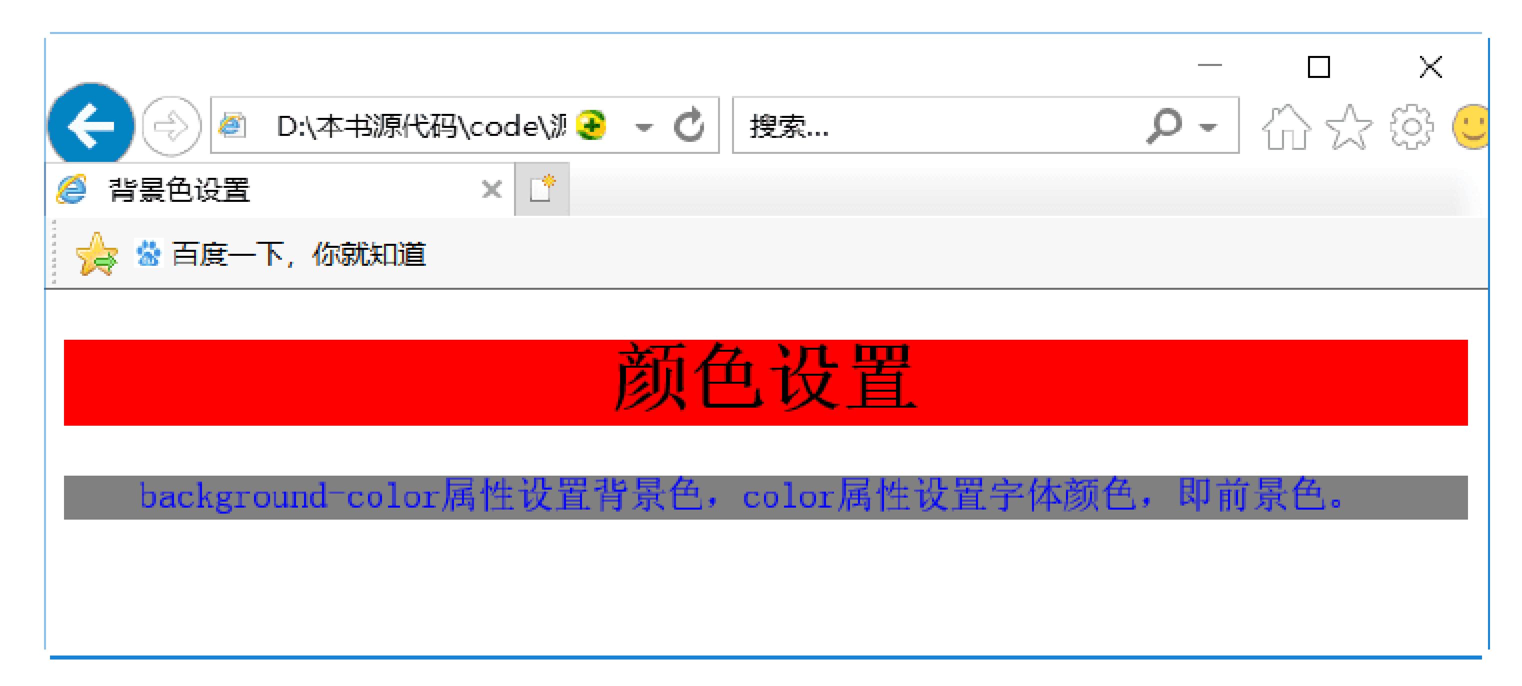


图 13-2　设置 HTML 元素背景色

13.1.2　背景图片

网页中不但可以使用背景色来填充网页背景，同样也可以使用背景图片来填充网页。通过 CSS3 属性可以对背景图片进行精确定位。background-image 属性用于设置标记的背景图片，通常在标记<body>中应用，将图片用于整个主体中。

background-image 语法格式如下：

```
background-image:none|url(url)
```

其默认属性为无背景图，当需要使用背景图时可以用 url 进行导入。url 可以使用绝对路径，也可以使用相对路径。

【例 13.3】（实例文件：ch13\13.3.html）

```
<!DOCTYPE html>
<html>
<head>
<title>背景色设置</title>
<style>
body{
    background-image:url(xiyang.jpg)
}
</style>
</head>
<body>
<p style="font-size:20pt">夕阳无限好</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-3 所示，可以看到网页中显示背景图，但如果图片大小小于整个网页大小，图片就会重复平铺整个网页。

在设置背景图片时，最好同时设置背景色，这样当背景图片因某种原因无法正常显示时，

可以使用背景色来代替。当然，如果正常显示，背景图片将覆盖背景色。

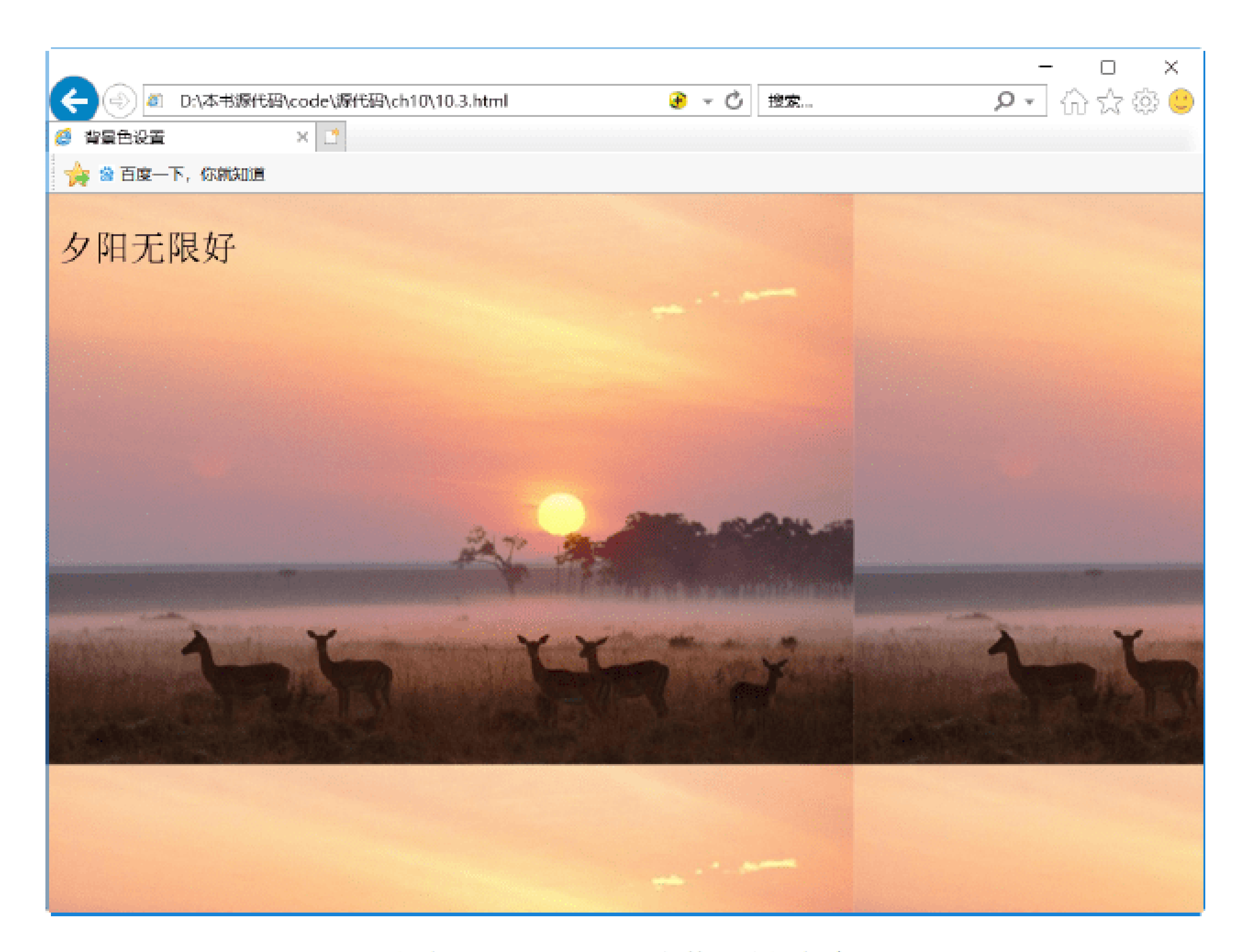


图 13-3　设置背景图片

13.1.3　背景图片重复

在进行网页设计时，通常都是一个网页使用一张背景图片，如果图片大小小于背景图片，则会直接重复平铺整个网页，但这种方式不适用于大多数页面。在 CSS 中可以通过 background-repeat 属性设置图片的重复方式，包括水平重复、垂直重复和不重复等。

background-repeat 属性用于设置背景图片是否重复平铺，各属性值说明如表 13-1 所示。

表 13-1　background-repeat 属性

属性值	描述
repeat	背景图片水平和垂直方向都重复平铺
repeat-x	背景图片水平方向重复平铺
repeat-y	背景图片垂直方向重复平铺
no-repeat	背景图片不重复平铺

background-repeat 属性重复背景图片是从元素的左上角开始平铺，直到水平、垂直或全部页面都被背景图片覆盖。

【例 13.4】（实例文件：ch13\13.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>背景图片重复</title>
```

```
<style>
body{
    background-image:url(xiyang.jpg); /*设置背景图片*/
    background-repeat:no-repeat;     /*设置背景图片不重复平铺*/
}
</style>
</head>
<body>
<p style="font-size:20pt">夕阳无限好</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-4 所示，可以看到网页中显示背景图，但图片以默认大小显示，而没有对整个网页背景进行填充，这是因为代码中设置了背景图不重复平铺。

同样可以在上面代码中设置 background-repeat 的属性值为其他值，如可以设置值为 repeat-x，表示图片在水平方向平铺。此时，在 IE 11.0 中浏览效果如图 13-5 所示。

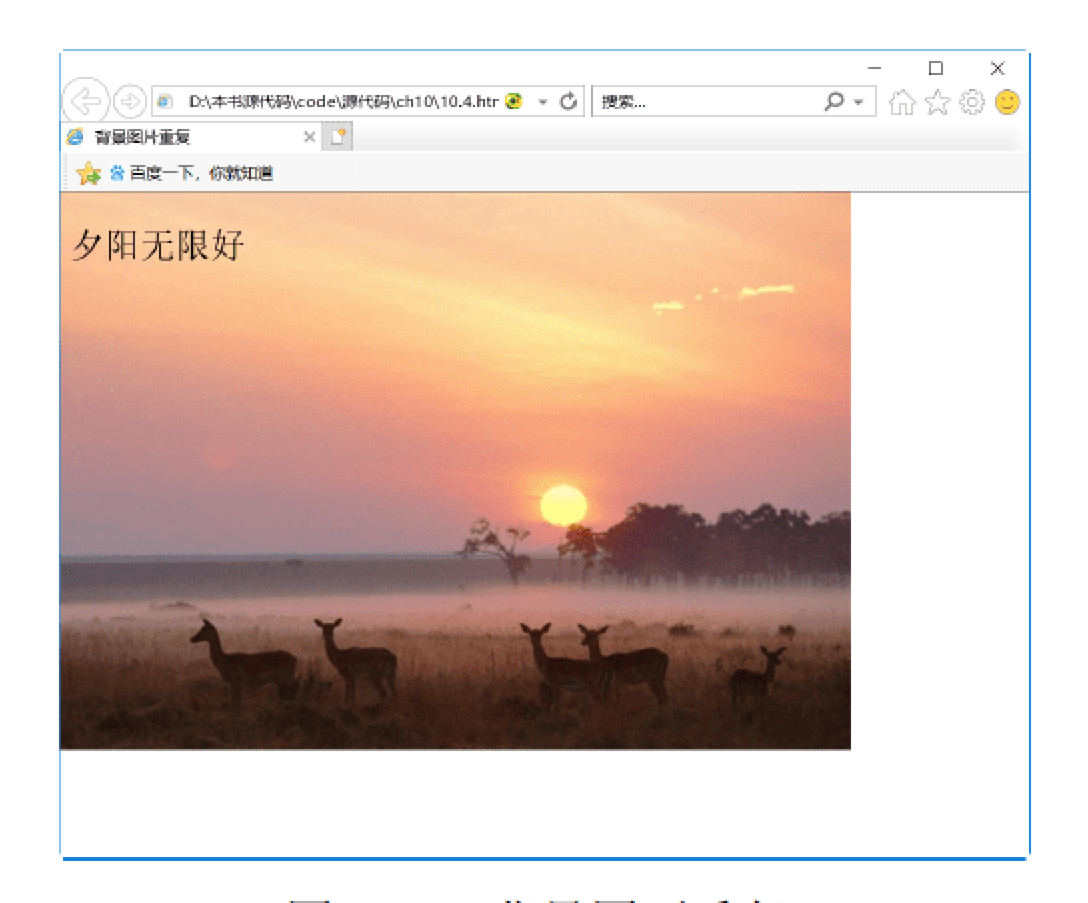


图 13-4　背景图不重复

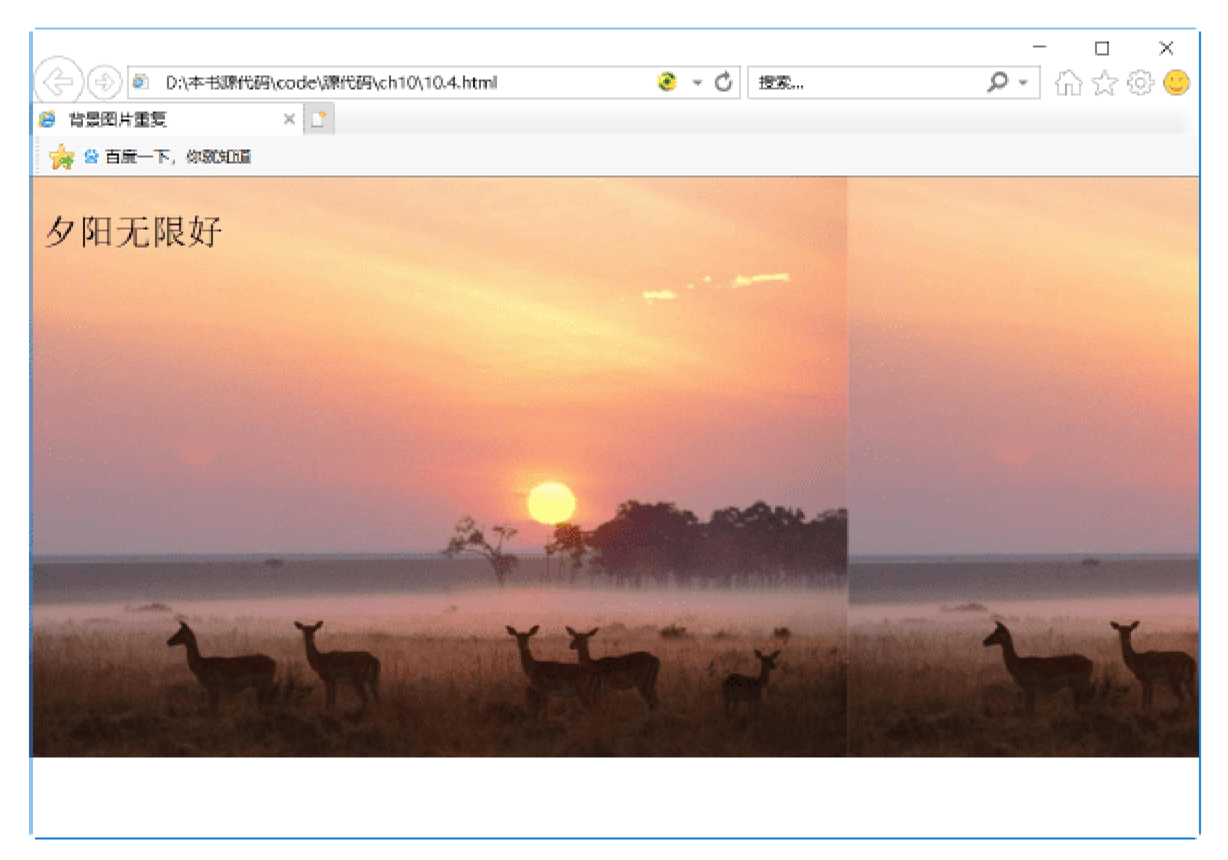


图 13-5　水平方向平铺

13.1.4　背景图片显示

对于一个文本较多，一屏显示不了的页面，如果使用的背景图片不足以覆盖整个页面，而且只将背景图片应用在页面的一个位置，那么在浏览页面时会出现看不到背景图片，或者背景图片初始可见，但随着页面的滚动又不可见的情况。也就是说，背景图片不能时刻随着页面的滚动显示。

要解决上述问题，用要使用 background-attachment 属性，该属性用来设置背景图片是否随文档一起滚动。该属性包含两个属性值：scroll 和 fixed，并适用于所有元素，如表 13-2 所示。

表 13-2　background-attachment 属性值

属性值	描述
scroll	默认值，当页面滚动时，背景图片随页面一起滚动
fixed	背景图片固定在页面的可见区域里

使用 background-attachment 属性可以使背景图片始终处于视野范围内，以避免出现因页面滚动而消失的情况。

【例 13.5】（实例文件：ch13\13.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>背景显示方式</title>
<style>
body{
    background-image:url(xiyang.jpg);     /*设置背景图片*/
    background-repeat:no-repeat;        /*设置背景图片不重复平铺*/
    background-attachment:fixed;       /*设置背景图片固定在可见区域里*/
}
p{
   text-indent:2em;                   /*设置文本缩进*/
line-height:30px;                 /*设置行高的大小*/
}
h1{text-align:center;}               /*设置标题居中显示*/
</style>
</head>
<body>
<h1>兰亭序</h1>
<p>永和九年，岁在癸（guǐ）丑，暮春之初，会于会稽（kuài jī）山阴之兰亭，修禊（xì）事也。群贤毕至，少长咸集。此地有崇山峻岭，茂林修竹， 又有清流激湍（tuān），映带左右。引以为流觞（shāng）曲（qū）水，列坐其次，虽无丝竹管弦之盛，一觞一咏，亦足以畅叙幽情。</p>
<p>是日也，天朗气清，惠风和畅。仰观宇宙之大，俯察品类之盛，所以游目骋（chěng）怀，足以极视听之娱，信可乐也。</p>
<p> 夫人之相与，俯仰一世。或取诸怀抱，晤言一室之内；或因寄所托，放浪形骸（hái）之外。虽趣（qǔ）舍万殊，静躁不同，当其欣于所遇，暂得于己，怏然自足，不知老之将至。及其所之既倦，情随事迁，感慨系（xì）之矣。向之所欣，俯仰之间，已为陈迹，犹不能不以之兴怀。况修短随化，终期于尽。古人云：“死生亦大矣。”岂不痛哉！</p>
<p>每览昔人兴感之由，若合一契，未尝不临文嗟（jiē）悼，不能喻之于怀。固知一死生为虚诞，齐彭殇(shāng)为妄作。后之视今，亦犹今之视昔，悲夫！故列叙时人，录其所述。虽世殊事异，所以兴怀，其致一也。后之览者，亦将有感于斯文。</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-6 所示，可以看到网页 background-attachment 属性的值为 fixed

时，背景图片的位置固定并不是相对于页面的，而是相对于页面的可视范围。

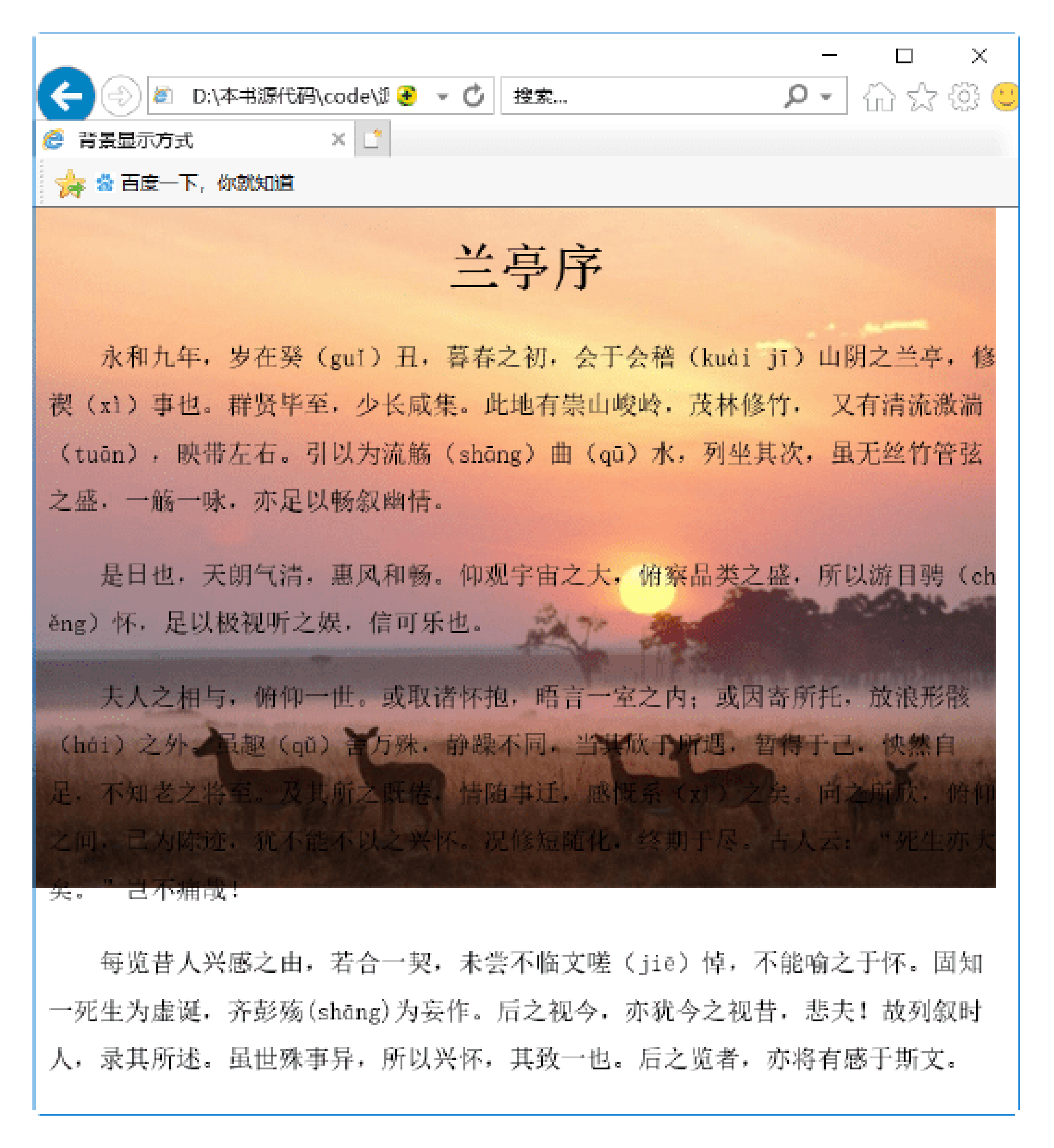


图 13-6　图片显示方式

13.1.5　背景图片位置

我们知道，背景图片都从设置了 background 属性网页的左上角开始出现，但在实际网页设计中，可以根据需要直接指定背景图片出现的位置。在 CSS3 中，可以通过 background-position 属性轻松调整背景图片的位置。

background-position 属性用于指定背景图片在页面中所处的位置。该属性值可以分为 4 类：绝对定义位置（length）、百分比定义位置（percentage）、垂直对齐值和水平对齐值。其中垂直对齐值包括 top、center 和 bottom，水平对齐值包括 left、center 和 right，如表 13-3 所示。

表 13-3　background-position 属性值

属性值	描述
length	设置图片与边距水平和垂直方向的距离长度，后跟长度单位（cm、mm、px 等）
percentage	以页面元素框的宽度或高度的百分比放置图片
top	背景图片顶部居中显示
center	背景图片居中显示
bottom	背景图片底部居中显示
left	背景图片左部居中显示
right	背景图片右部居中显示

垂直对齐值还可以与水平对齐值一起使用，从而决定图片的垂直位置和水平位置。

【例 13.6】（实例文件：ch13\13.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>背景位置设置</title>
<style>
body{
    background-image:url(xiyang.jpg);  /*设置背景图片*/
    background-repeat:no-repeat;       /*设置背景图片不重复平铺*/
    background-position:top right;     /*设置背景图片从顶部和右边开始显示*/
}
</style>
</head><body></body></html>
```

在 IE 11.0 中浏览效果如图 13-7 所示，可以看到网页中显示背景，其背景是从顶部和右边开始出现的。

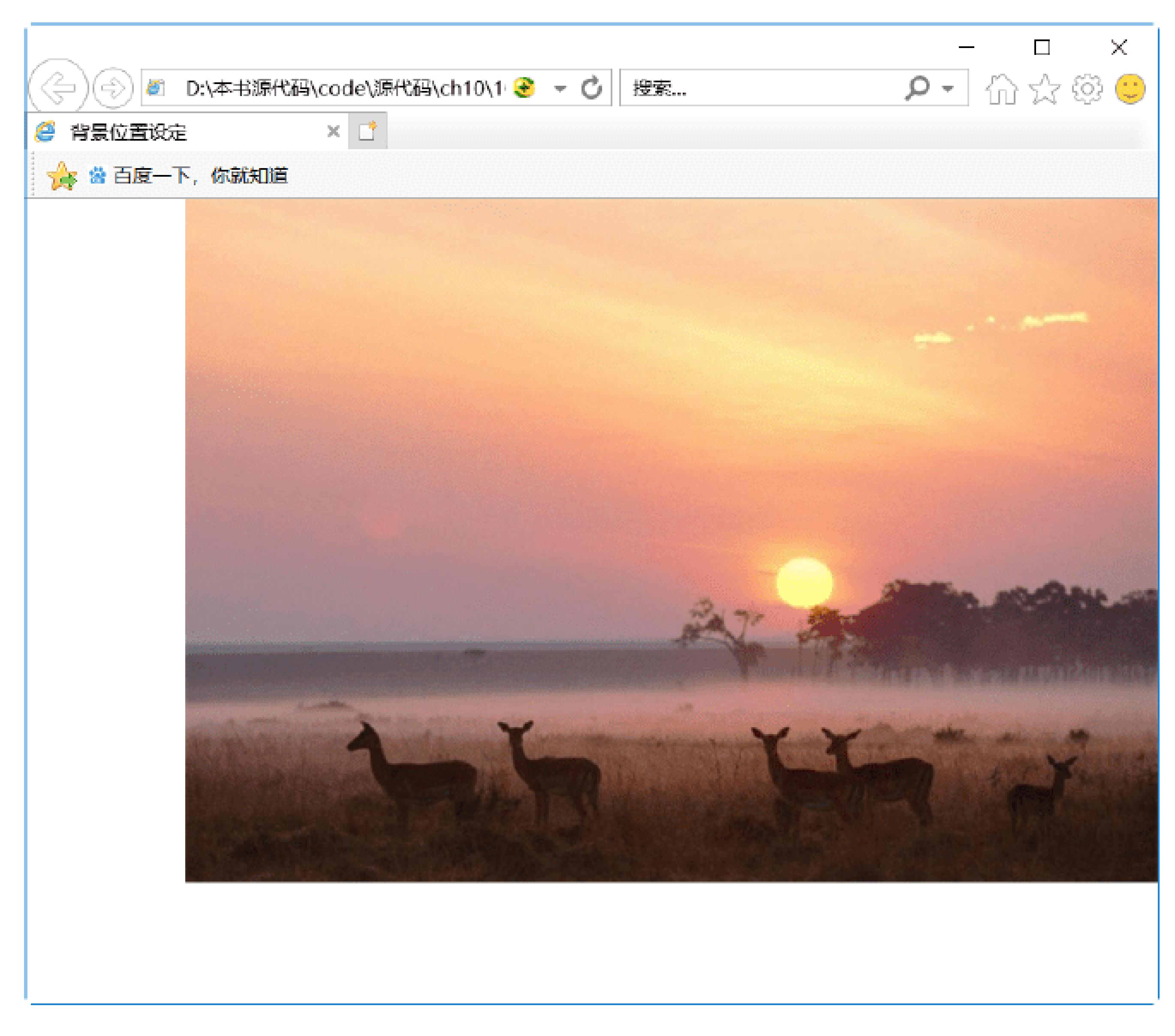


图 13-7　设置背景位置

使用垂直对齐值和水平对齐值只能格式化地放置图片，如果要在页面中自由地定义图片的位置，则需要使用确定数值或百分比。此时需将上面代码中：

```
background-position:top right;
```

语句修改为：

```
background-position:20px 30px
```

在 IE 11.0 中浏览效果如图 13-8 所示，可以看到网页中显示背景，其背景是从左上角开始出现的，开始位置坐标为（20，30）。

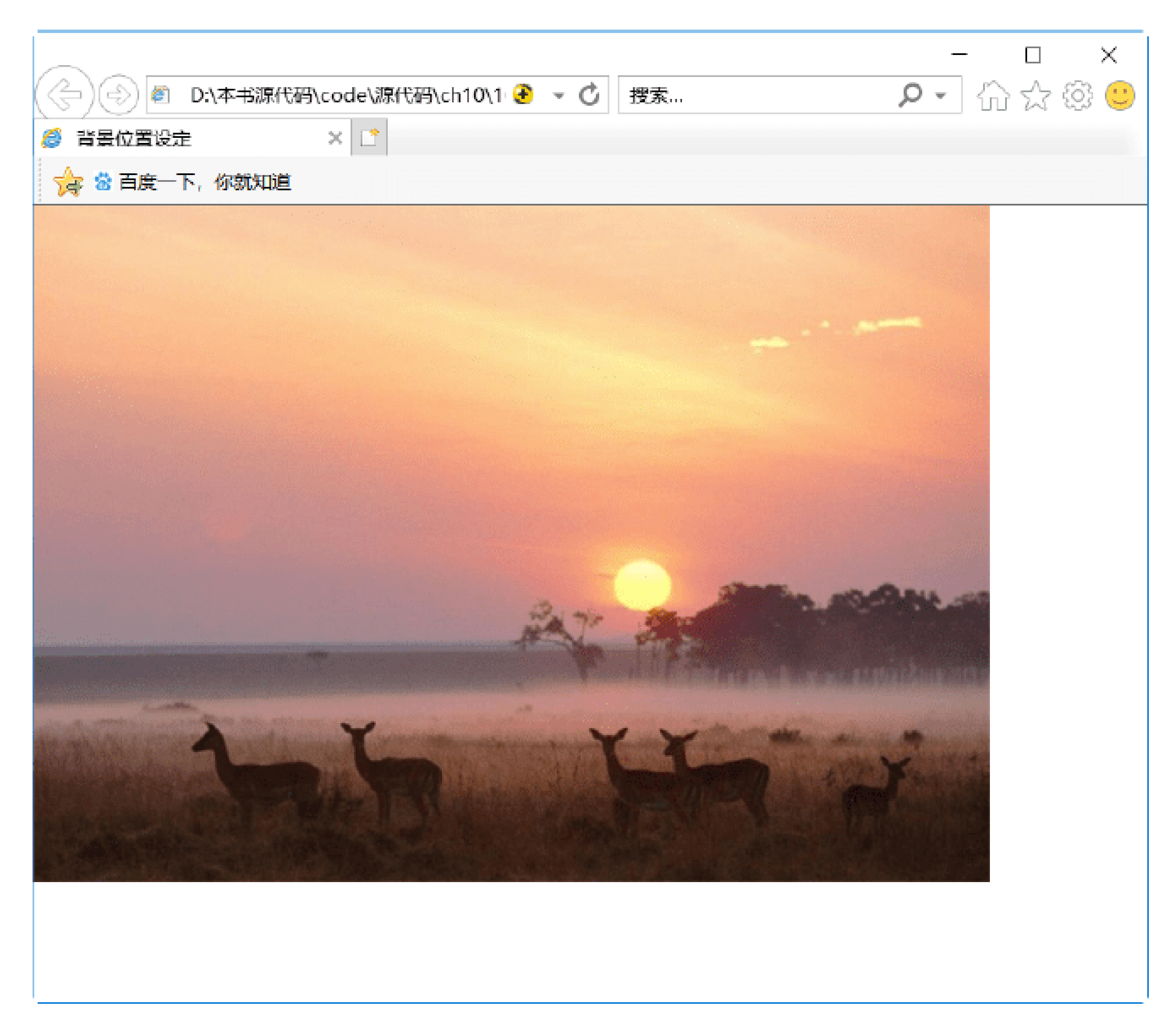


图 13-8　指定背景位置

13.1.6　背景图片大小

在以前的网页设计中，背景图片的大小是不可以控制的，如果想要图片填充整个背景，则需要事先设计一个较大的背景图片，要么只能让背景图片以平铺的方式来填充页面元素。在 CSS3 中，新增了一个 background-size 属性，用来控制背景图片大小，从而降低网页设计的开发成本。

background-size 语法格式如下：

```
background-size: [<length>|<percentage>|auto]{1,2}|cover|contain
```

其参数值含义如表 13-4 所示。

表 13-4　background-size 属性参数表

参数值	说明
<length>	由浮点数字和单位标识符组成的长度值，不可为负值
<percentage>	取值为 0%~100%，不可为负值
cover	保持背景图像本身的宽高比例，将图片缩放到正好完全覆盖所定义的背景区域
contain	保持图像本身的宽高比较，将图片缩放到宽度或高度正好适应所定义的背景区域

【例 13.7】（实例文件：ch13\13.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title>背景位置设置</title>
<style>
body{
    background-image:url(xiyang.jpg); /*设置背景图片*/
    background-repeat:no-repeat;      /*设置背景图片不重复平铺*/
    background-size:cover;            /*设置背景图片填满了整个页面*/
}
</style>
</head>
<body>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-9 所示，可以看到网页中的背景图片填充了整个页面。

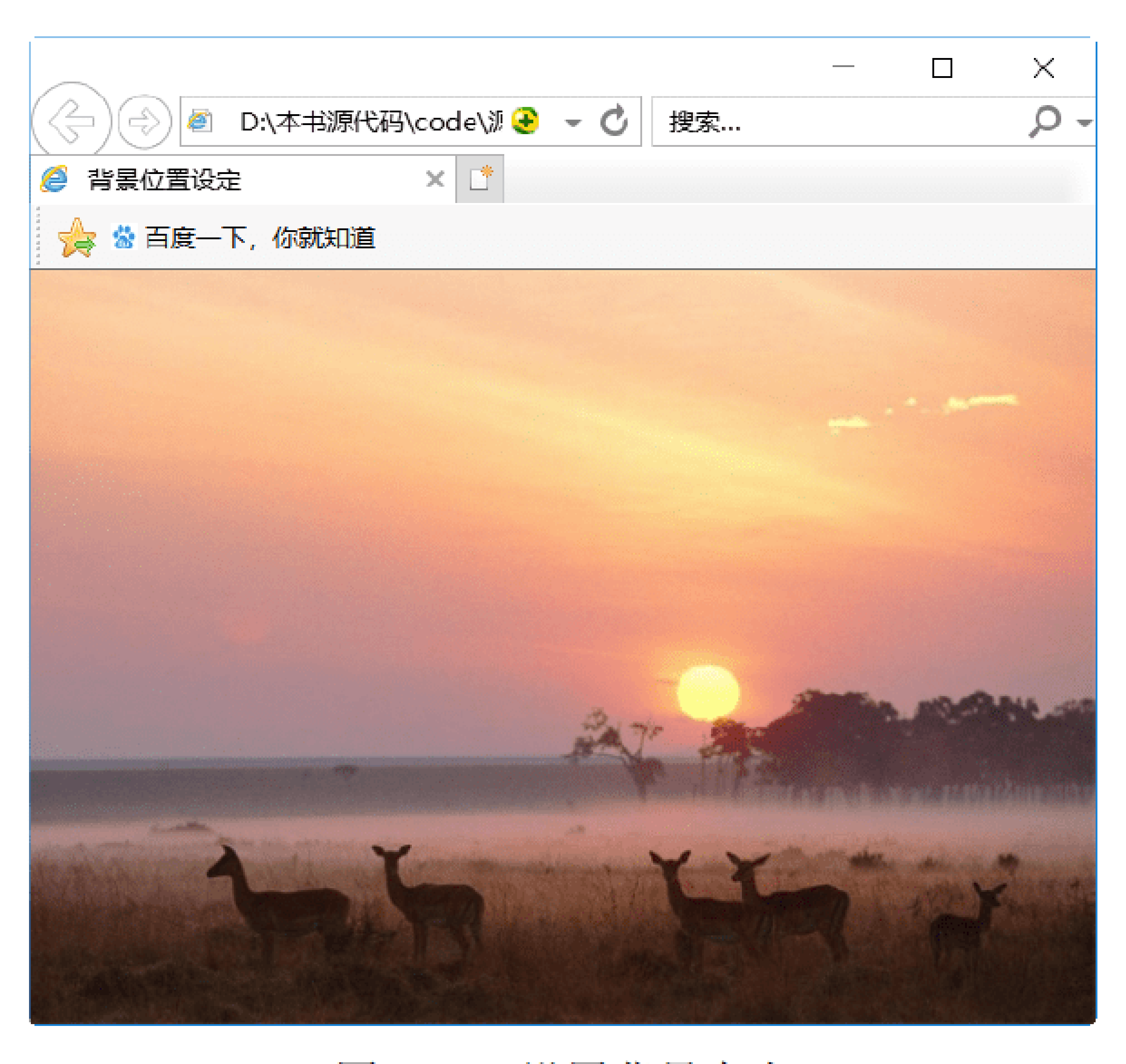


图 13-9　设置背景大小

同样也可以用像素或百分比指定背景大小显示。当指定为百分比时，大小会由所在区域的宽度、高度及 background-origin 的位置决定。示例如下：

```
background-size:900 800;
```

此时 background-size 属性可以设置 1 个或 2 个值，1 个为必填，1 个为选填。其中第 1 个值用于指导图片宽度，第 2 个值用于指定图片高度，如果只设置一个值，则第 2 个值默认为 auto。

13.1.7 背景显示区域

在网页设计中，如果能改善背景图片的定位方式，使设计师能够更加灵活地决定背景图片应该显示的位置，则会大大减少设计成本。在 CSS3 中，新增了一个 background-origin 属性，用来完成背景图片的定位。

默认情况下，background-position 属性总是以元素左上角原点作为背景图像定位，而 background-origin 属性可以改变这种定位方式。

```
background-origin: border|padding|content
```

其参数含义如表 13-5 所示。

表 13-5 background-origin 参数值表

参数值	说明
border	从 border 区域开始显示背景
padding	从 padding 区域开始显示背景
content	从 content 区域开始显示背景

【例 13.8】（实例文件：ch13\13.8.html）

```
<!DOCTYPE html>
<html>
<head>
<title>背景坐标原点</title>
<style>
div{
    text-align:center;              /*设置居中显示*/
    height:500px;                   /*设置div块的高度*/
    width:416px;                    /*设置div块的宽度*/
    border:solid 1px red;           /*设置div块的边框样式*/
    padding:32px 2em 0;             /*设置内边距的大小*/
    background-image:url(15.jpg);/*设置背景图片*/
    background-origin:padding;      /*设置从padding区域开始显示背景*/
}
div h1{
   font-size:18px;
   font-family:"幼圆";
}
div  p{
   text-indent:2em;
   line-height:2em;
   font-family:"楷体";
}
</style>
</head>
<body>
<div>
<h1>美科学家发明时光斗篷 在时间中隐瞒事件</h1>
<p>本报讯据美国《技术评论》杂志网站7月15日报道，日前，康奈尔大学的莫蒂·弗里德曼和其
```

```
同事在前人研究的基础上，设计并制造出了一种能在时间中隐瞒事件的时光斗篷。相关论文发表在国际著名学术网站arXiv.org上。</p>
    <p>近年来有关隐身斗篷的研究不断取得突破，其原理是通过特殊的材料使途经的光线发生扭曲，从而让斗篷下的物体“隐于无形”。第一个隐身斗篷只在微波中才有效果，但短短几年，物理学家已经发明出了能用于可见光的隐身斗篷，能够隐藏声音的“隐声斗篷”和能让一个物体看起来像其他物体的“错觉斗篷”。</p>
    </div>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 13-10 所示，可以看到背景图片以指定大小于网页左侧显示，在背景图片上显示了相应的段落信息。

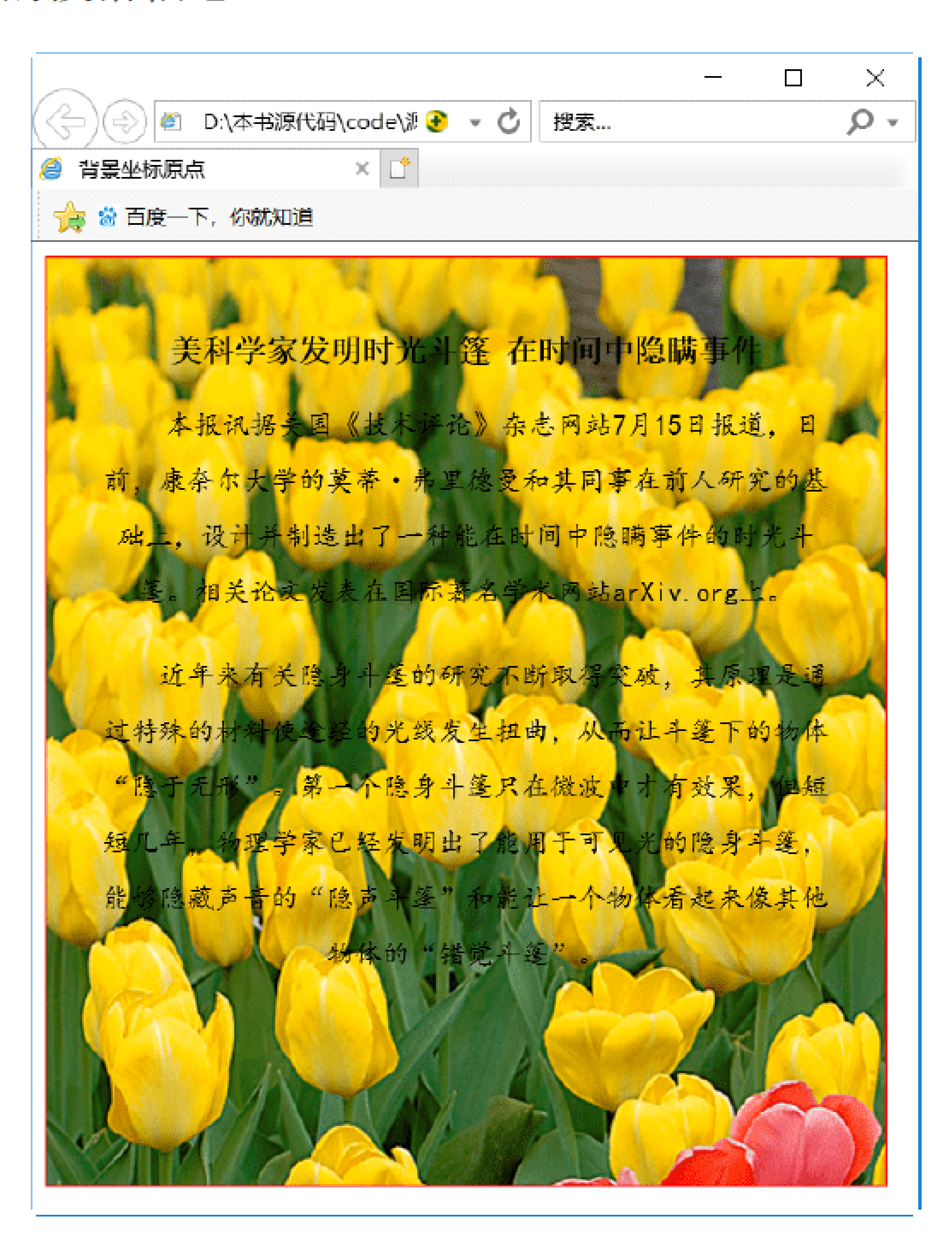


图 13-10　设置背景显示区域

13.1.8　背景图像裁剪区域

在 CSS3 中，新增了一个 background-clip 属性，用来定义背景图片的裁剪区域。background-clip 属性和 background-orgin 属性有几分相似，通俗地说，background-clip 属性用来判断背景是否包含边框区域，而 background-orgin 属性用来决定 background-position 属性定位的参考位置。

background-clip 语法格式如下：

```
background-clip: border-box|padding-box|content-box|no-clip
```

其参数值含义如表 13-6 所示。

表 13-6　background-clip 参数值表

参数值	说明	参数值	说明
border	从 border 区域开始显示背景	content	从 content 区域开始显示背景
padding	从 padding 区域开始显示背景	no-clip	从边框区域外裁剪背景

【例 13.9】（实例文件：ch13\13.9.html）

```
<!DOCTYPE html>
<html>
<head>
<title>背景裁剪</title>
<style>
div{
    height:300px;                          /*设置div块的高度*/
    width:200px;                           /*设置div块的宽度*/
    border:dotted 50px red;                /*设置边框的样式*/
    padding:50px;                          /*设置内边距的大小*/
    background-image:url(18.jpg);          /*设置背景图片*/
    background-repeat:no-repeat;           /*设置背景图片不重复平铺*/
    background-clip:content;}              /* 背景图像仅在内容区域中显示*/
</style>
</head>
<body>
<div>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-11 所示，可以看到网页中背景图像仅在内容区域内显示。

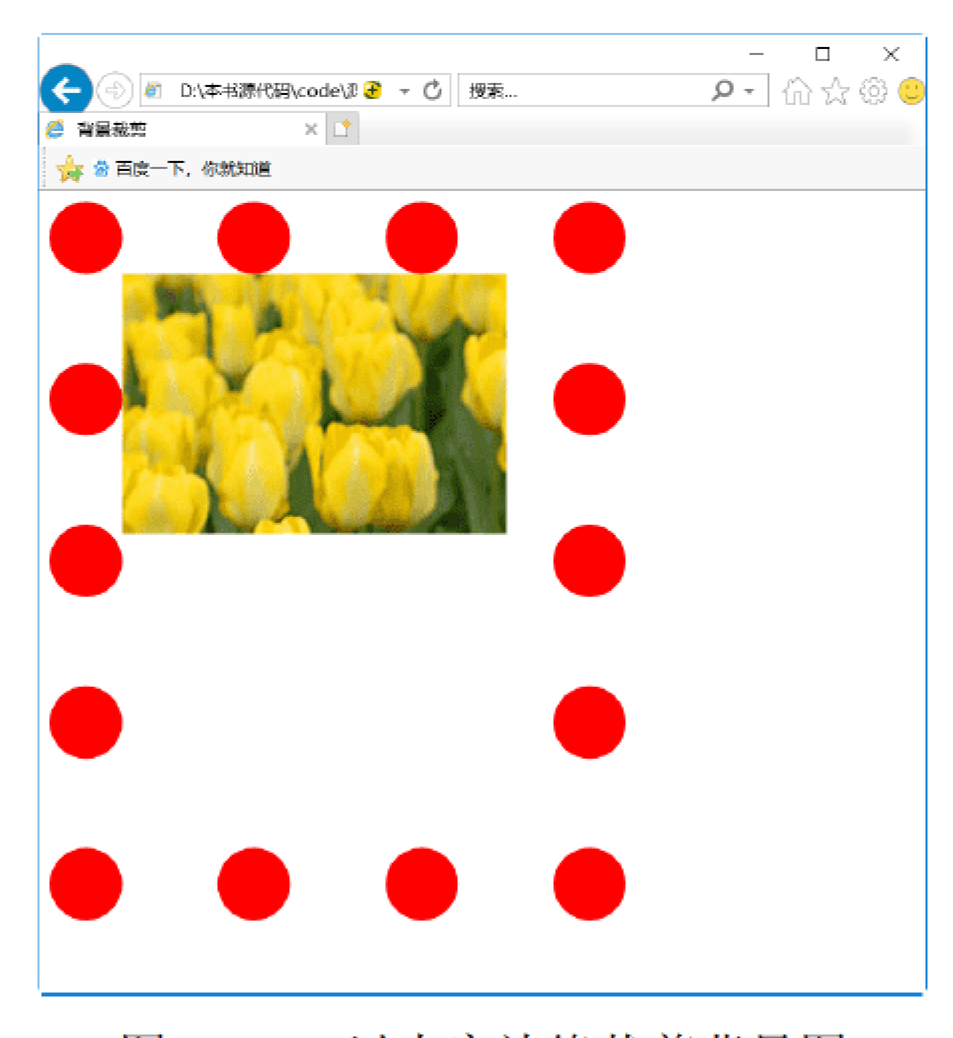


图 13-11　以内容边缘裁剪背景图

13.1.9 背景复合属性

在 CSS3 中，background 属性依然保持了一起的用法，即综合以上所有与背景有关的属性（即以 back-ground-开头的属性）一次性地设置背景样式。格式如下：

```
background:[background-color] [background-image] [background-repeat]
      [background-attachment] [background-position]
 [background-size] [background-clip] [background-origin]
```

其中的属性顺序可以自由调换并且可以选择设置，对于没有设置的属性系统会自行为该属性添加默认值。例如，定义背景样式规则如下：

```
body
{
  background-color:black;          /*设置背景颜色*/
  background-image:url(bk1.jpg); /*设置背景图片*/
  background-position:center;     /*设置背景图片居中显示*/
  background-repeat:repeat-x;     /*设置背景图片水平平铺*/
  background-attachment:fixed;    /*设置背景图片固定在可见区域里*/
  background-size:900 800;        /*设置背景图片的大小*/
  background-origin:padding;      /*设置从padding区域开始显示背景*/
  background-clip:content;       /*从content开始剪切背景图片*/
}
```

13.2 边框

在网页设计中，HTML 元素会占有一定的区域，如<a>标记、<p>标记等，对于这些标记可以通过 CSS3 的 width 和 height 设置其大小，并通过 border 设置其边框样式。

边框就是将元素内容及间隙包含在其中的边线，类似于表格的外边线。每一个页面元素的边框可以从 3 个方面来描述：宽度、样式和颜色，这 3 个方面决定了边框所显示出来的外观。CSS3 中分别使用 border-style、border-width 和 border-color 3 个属性设置边框的 3 个方面。

13.2.1 边框样式

在进行网页排版时，有时需要指定某个区域的元素，并将这些元素与其他元素区别开来，这时可以让 HTML 元素带有边框并设置 HTML 边框样式。

border-style 属性用于设置边框的样式，也就是风格。设置边框格式是边框最重要的部分，它主要用于为页面元素添加边框。其语法格式如下：

```
border-style:none|hidden|dotted|dashed|solid|double|groove|ridge|inset|
outset
```

CSS3 设置了 9 种边框样式，如表 13-7 所示。

表 13-7　边框样式属性值

属性值	描述
none	无边框，无论边框宽度设为多大
dotted	点线式边框
dashed	破折线式边框
solid	直线式边框
double	双线式边框
groove	槽线式边框
ridge	脊线式边框
inset	内嵌效果的边框
outset	突起效果的边框

【例 13.10】（实例文件：ch13\13.10.html）

```
<!DOCTYPE html>
<html>
<head>
<title>边框设置</title>
<style>
h1{
   border-style:dotted;     /*点线式边框*/
   color: black;            /*设置标题颜色*/
   text-align:center;       /*设置标题居中显示*/
}
p{
   border-style:double;     /*双线式边框*/
   text-indent:2em;         /*设置文本缩进*/
}
</style>
</head>
<body>
<h1>带有边框的标题</h1>
<p>带有边框的段落</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-12 所示，可以看到网页中标题 h1 显示的时候带有边框，其边框样式为点线式边框；同样段落也带有边框，其边框样式为双线式边框。



在没有设置边框颜色的情况下，groove、ridge、inset 和 outset 边框默认的颜色是灰色；dotted、dashed、solid 和 double 这 4 种边框的颜色基于页面元素的 color 值。

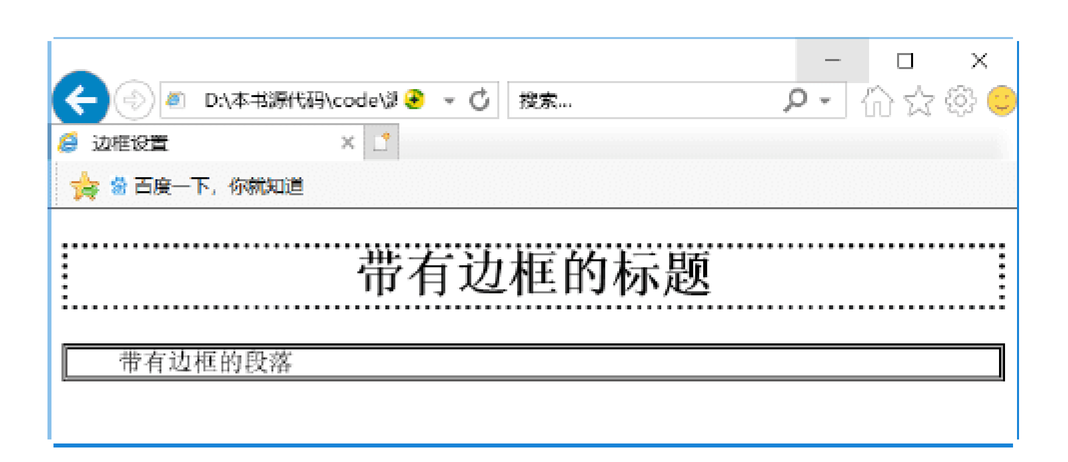


图 13-12　设置边框

其实，这几种边框样式还可以分别定义在一个边框中，从上边框开始按照顺时针的方向分别定义边框的上、右、下、左边框样式，从而形成多样式边框。例如，有下面一条样式规则：

```
p{border-style:dotted solid dashed groove}
```

另外，如果需要单独定义边框的一条边的样式，则可以使用如表 13-8 所列的属性来定义。

表 13-8　边样式定义属性

属性	描述
border-top-style	设置上边框的样式
border-right-style	设置右边框的样式
border-bottom-style	设置下边框的样式
border-left-style	设置左边框的样式

13.2.2　边框颜色

在网页设计中，设计者不但可以设置边框样式，还可以设置边框颜色，从而增强边框的效果。border-color 属性用于设置边框颜色，如果不想与页面元素的颜色相同，则可以使用该属性为边框定义其他颜色。

border-color 属性语法格式如下：

```
border-color:color
```

color 表示指定颜色，其颜色值通过十六进制或 RGB 等方式获取。

与边框样式属性一样，border-color 属性可以为边框设置一种颜色，也可以同时设置 4 个边的颜色。

【例 13.11】（实例文件：ch13\13.11.html）

```
<!DOCTYPE html>
<html>
<head>
<title>边框颜色设置</title>
<style>
p{
   border-style:double;
   border-color:red;
   text-indent:2em;
}
</style>
</head>
<body>
<p>边框颜色设置</p>
<p style="border-style:solid; border-color:red blue yellow green">
分别定义边框颜色</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-13 所示，可以看到网页中第一段落边框颜色设置为红色，第二段落边框颜色分别设置为红、蓝、黄和绿。

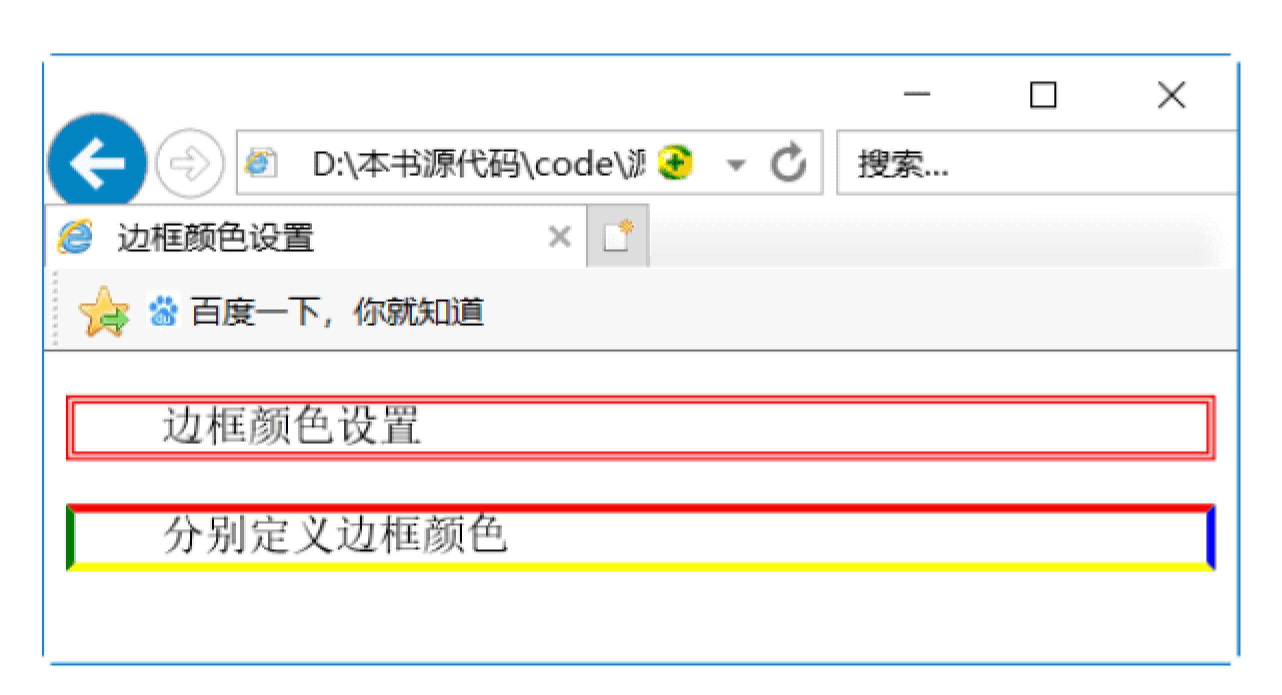


图 13-13　设置边框颜色

除了上面设置 4 个边框颜色的方法之外，还可以使用表 13-9 列出的属性单独为相应的边框设置颜色。

表 13-9　相应的边框颜色设置属性

属性	描述
border-top-color	设置上边框颜色
border-right-color	设置右边框颜色
border-bottom-color	设置下边框颜色
border-left-color	设置左边框颜色

13.2.3　边框线宽

在 CSS3 中，可以通过设置边框线宽来增强边框效果。border-width 属性可以用来设置边框宽度，其语法格式如下：

```
border-width:medium|thin|thick|length
```

其中预设有 3 种属性值：medium、thin 和 thick。还可以自行设置宽度（width），如表 13-10 所示。

表 13-10　边框线宽属性值

属性值	描述
medium	默认值，中等宽度
Thin	比 medium 细
thick	比 medium 粗
length	自定义宽度

【例 13.12】（实例文件：ch13\13.12.html）

```
<!DOCTYPE html>
<html>
<head>
<title>边框宽度设置</title>
</head>
<body>
   <p style="border-style:dotted; border-width:medium;">边框颜色设置</p>
   <p style="border-style:dashed;border-width:thin;">边框颜色设置</p>
   <p style="border-style:solid; border-width:12px;">分别定义边框颜色</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-14 所示，可以看到网页中 3 个段落边框以不同的粗细显示。

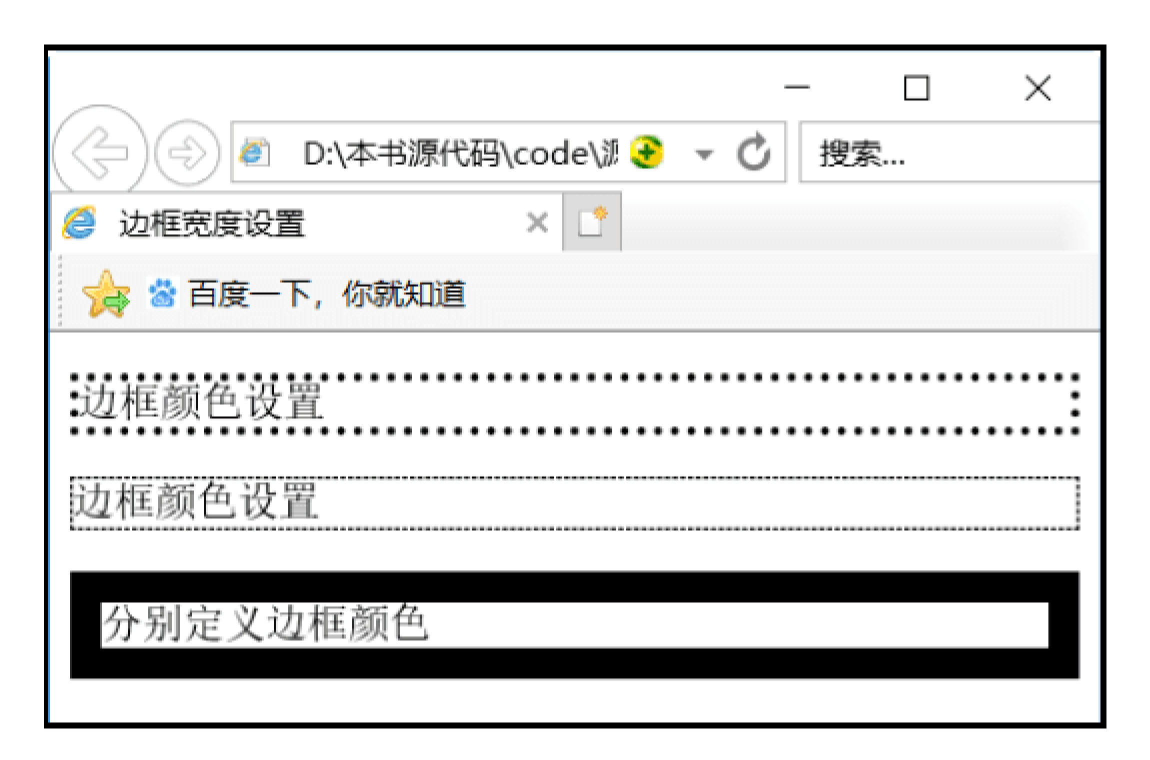


图 13-14　设置边框线宽

border-width 属性其实是 border-top-width、border-right-width、border-bottom-width 和 border-left- width 4 个属性的综合属性，分别用于设置上边框、右边框、下边框、左边框的宽度。

【例 13.13】（实例文件：ch13\13.13.html）

```
<!DOCTYPE html>
<html>
<head>
<title>边框宽度设置</title>
<style>
p{
border-style:solid;       /*直线式边框*/
border-color:#ff00ee;      /*设置边框的颜色*/
border-top-width:medium;  /*定义中等的上边框*/
border-right-width:thin;  /*定义细的右边框*/
bottom-width:thick;       /*定义粗的下边框*/
border-left-width:15px;  /*定义左边框的粗细*/
}
</style>
</head>
<body>
<p>边框宽度设置</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-15 所示，可以看到网页中段落的 4 个边框以不同的宽度显示。

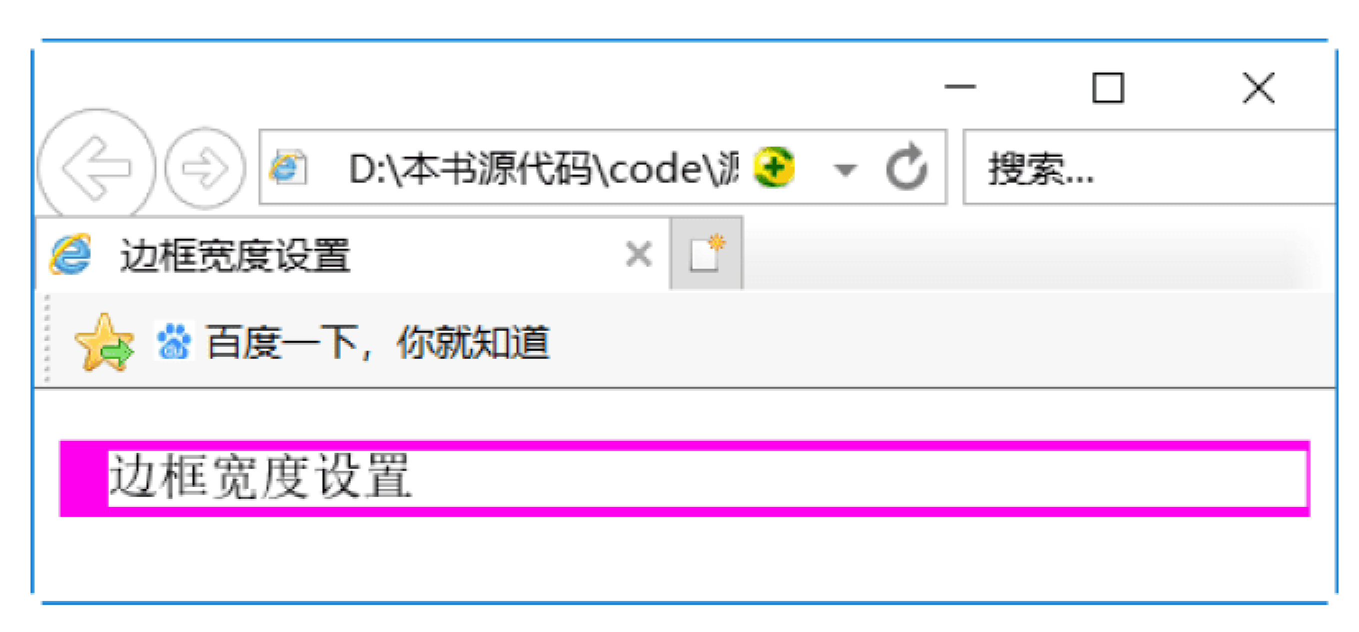


图 13-15　分别设置 4 个边框宽度

13.2.4　边框复合属性

border 属性集合了上述所介绍的 3 种属性，为页面元素设置边框的宽度、样式和颜色。语法格式如下：

```
border:border-width border-style border-color
```

其中，3 个属性顺序可以自由调换。

【例 13.14】（实例文件：ch13\13.14.html）

```
<!DOCTYPE html>
<html>
<head>
<title>边框复合属性设置</title>
</head>
<body>
<p style="border:dashed  red 12px">边框复合属性设置</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-16 所示，可以看到网页中段落边框样式以破折线显示，颜色为红色，线宽为 12 像素。

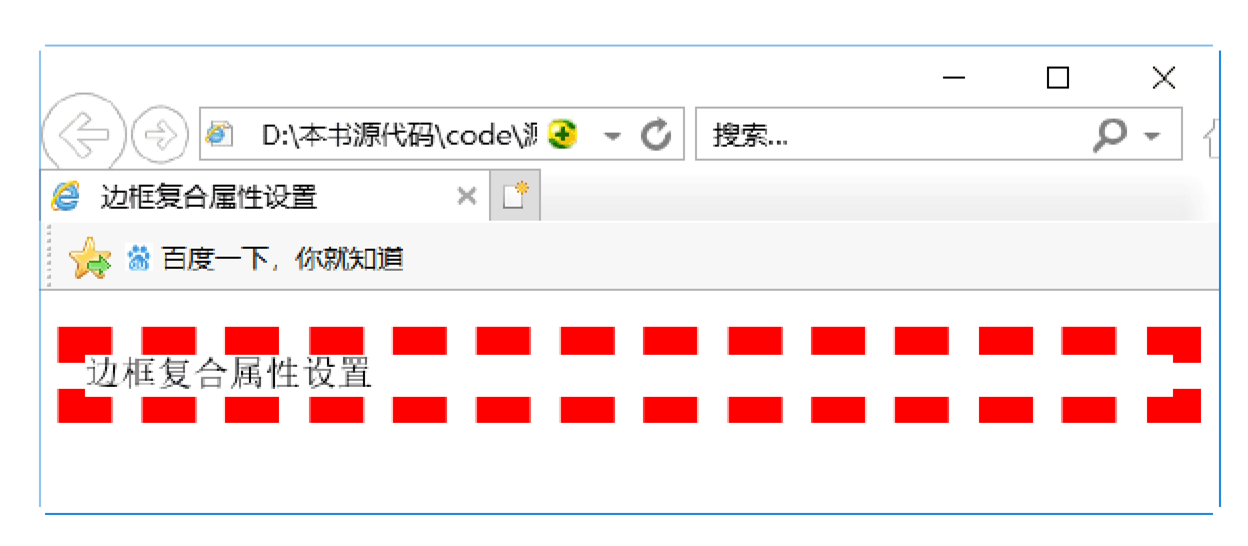


图 13-16　边框复合属性设置

13.3　圆角边框

在 CSS3 标准没有指定之前，如果想要实现圆角效果，则需要花费很多时间去实现。一方面需要照顾大多数的低版本 IE 用户，另一方面还需要兼容各种浏览器的私有属性。在 CSS3 标准推出后，网页设计者可以使用 border-radius 属性轻松实现圆角效果。

13.3.1　圆角边框属性

在 CSS3 中，可以使用 border-radius 属性定义边框的圆角效果，从而大大降低了圆角开发的成本。border-radius 的语法格式如下：

```
border-radius: none|<length>{1,4} [/<length>{1,4}]?
```

其中，none 为默认值，表示元素没有圆角；<length>表示由浮点数字和单位标识符组成的长度值，不可为负值。

【例 13.15】（实例文件：ch13\13.15.html）

```
<!DOCTYPE html>
<html>
```

```
<head>
<title>圆角边框设置</title>
<style>
p{text-align:center;
   border:15px solid red;    /*设置边框的样式*/
   width:100px;
   height:50px;
   border-radius:10px;      /*设置圆角边框的半径*/
}
</style>
</head>
<body>
   <p>这是一个圆角边框</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-17 所示，可以看到网页中段落边框显示时以圆角显示，其半径为 10 像素。

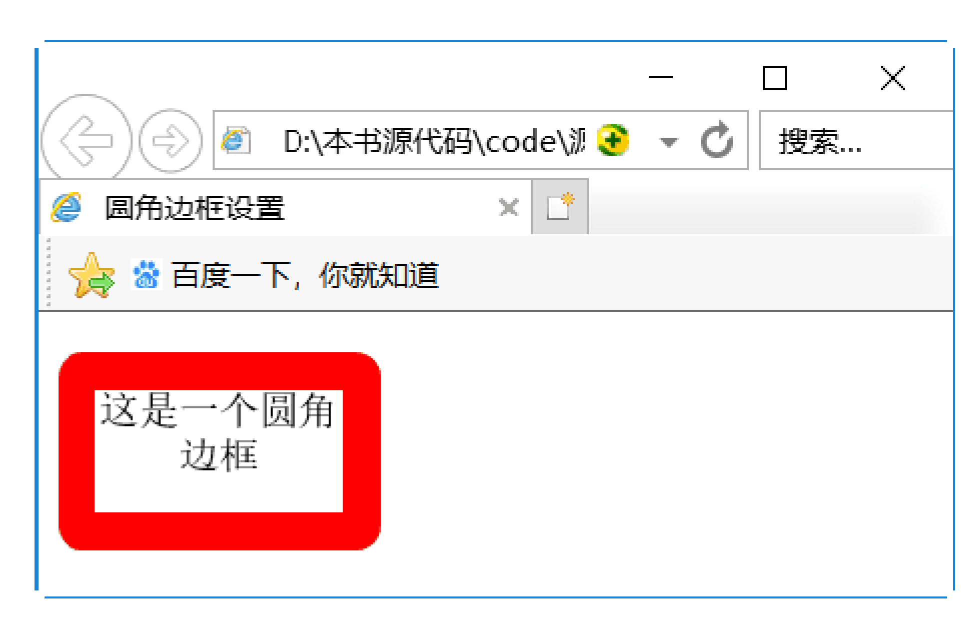


图 13-17　定义圆角边框

13.3.2　指定两个圆角半径

可以使用 border-radius 属性设置一个参数来绘制圆角，同样也可以使用两个参数绘来制圆角。border-radius 属性可以包含两个参数值：第一个参数表示圆角的水平半径；第二个参数表示圆角的垂直半径，两个参数通过斜线（“/”）隔开。如果仅含一个参数值，则第二个值与第一个值相同，表示这个角就是一个 1/4 的圆；如果参数值中包含 0，则这个值就是矩形，不会显示为圆角。

【例 13.16】（实例文件：ch13\13.16.html）

```
<!DOCTYPE html>
<html>
<head>
<title>圆角边框设置</title>
<style>
.p1{
```

```
        text-align:center;
        border:15px solid red;    /*设置边框的样式*/
        width:100px;
        height:50px;
        border-radius:5px/50px;   /*增加圆角边框*/
    }
    .p2{
        text-align:center;
        border:15px solid red;    /*设置边框的样式*/
        width:100px;
        height:50px;
        border-radius:50px/5px;   /*增加圆角边框*/
    }
    </style>
    </head>
    <body>
        <p class=p1>这是一个圆角边框A</p>
        <p class=p2>这也是一个圆角边框B</p>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 13-18 所示，可以看到网页中显示了两个圆角边框，第一段落圆角半径为 5 像素/50 像素，第二段落圆角半径为 50 像素/5 像素。

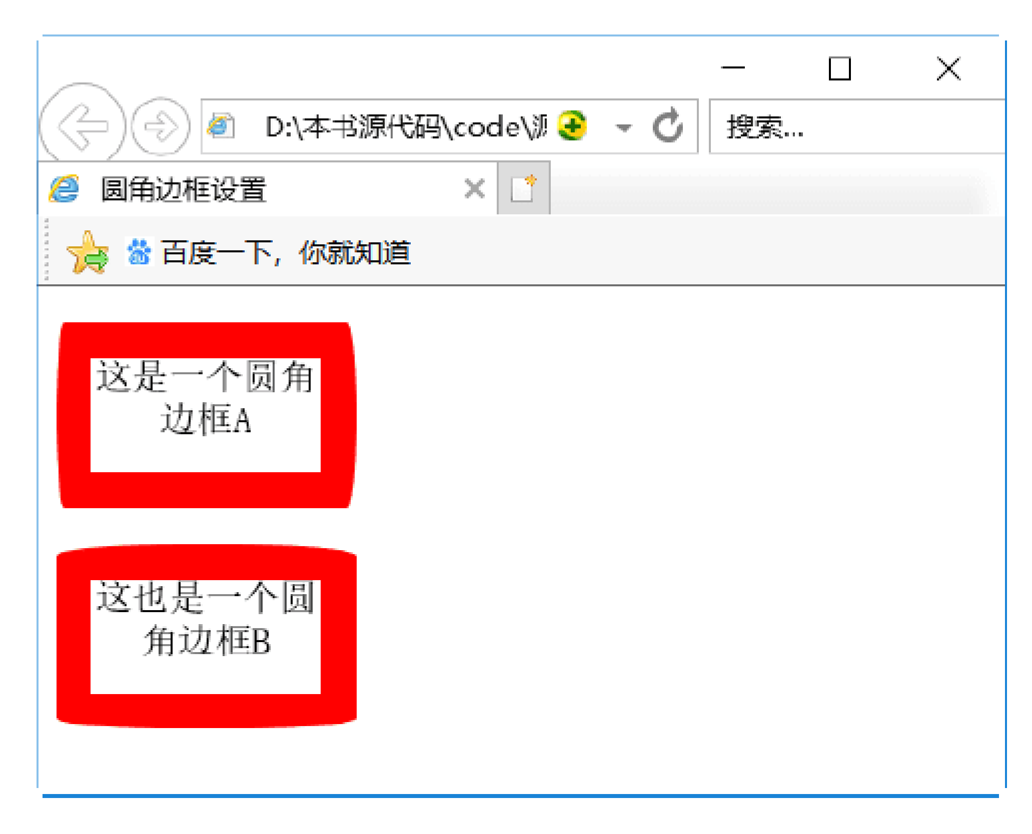


图 13-18　定义不同半径圆角边框

13.3.3　绘制 4 个不同圆角边框

有时为了网页需要，可以为圆角边框设置不同半径的圆角，其样式就会更加漂亮。在 CSS3 中，实现 4 个不同圆角边框有两种方法：一种是使用 border-radius 属性；另一种是使用 border-radius 衍生属性。

1. border-radius 属性

第一种方法就是利用 border-radius 属性为其赋一组值。当为 border-radius 属性赋一组值时

将遵循 CSS 规则，可以包含 2~4 个属性值，此时无法使用斜杠定义圆角水平和垂直半径。

如果直接给 border-radius 属性赋 4 个值，那么这 4 个值将按照 top-left、top-right、bottom-right、bottom-left 的顺序来设置。如果 bottom-left 值省略，则其圆角效果和 top-right 效果相同；如果 bottom-right 值省略，则其圆角效果和 top-left 效果相同；如果 top-right 的值省略，则其圆角效果和 top-left 效果相同。如果为 border-radius 属性设置 4 个值的集合参数，则每个值表示每个角的圆角半径。

【例 13.17】（实例文件：ch13\13.17.html）

```
<!DOCTYPE html>
<html>
<head>
<title>圆角边框设置</title>
<style>
.div1{
   border:15px solid blue;
   height:100px;
   border-radius:10px 30px 50px 70px;
}
.div2{
   border:15px solid blue;
   height:100px;
   border-radius:10px 50px 70px;
}
.div3{
   border:15px solid blue;
   height:100px;
   border-radius:10px 50px;
}
</style>
</head>
<body>
<div class=div1></div><br />
<div class=div2></div><br />
<div class=div3></div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-19 示，可以看到网页中第一个 div 层设置了 4 个不同的圆角边框，第二个 div 层设置了 3 个不同的圆角边框，第三个 div 层设置了两个不同的圆角边框。

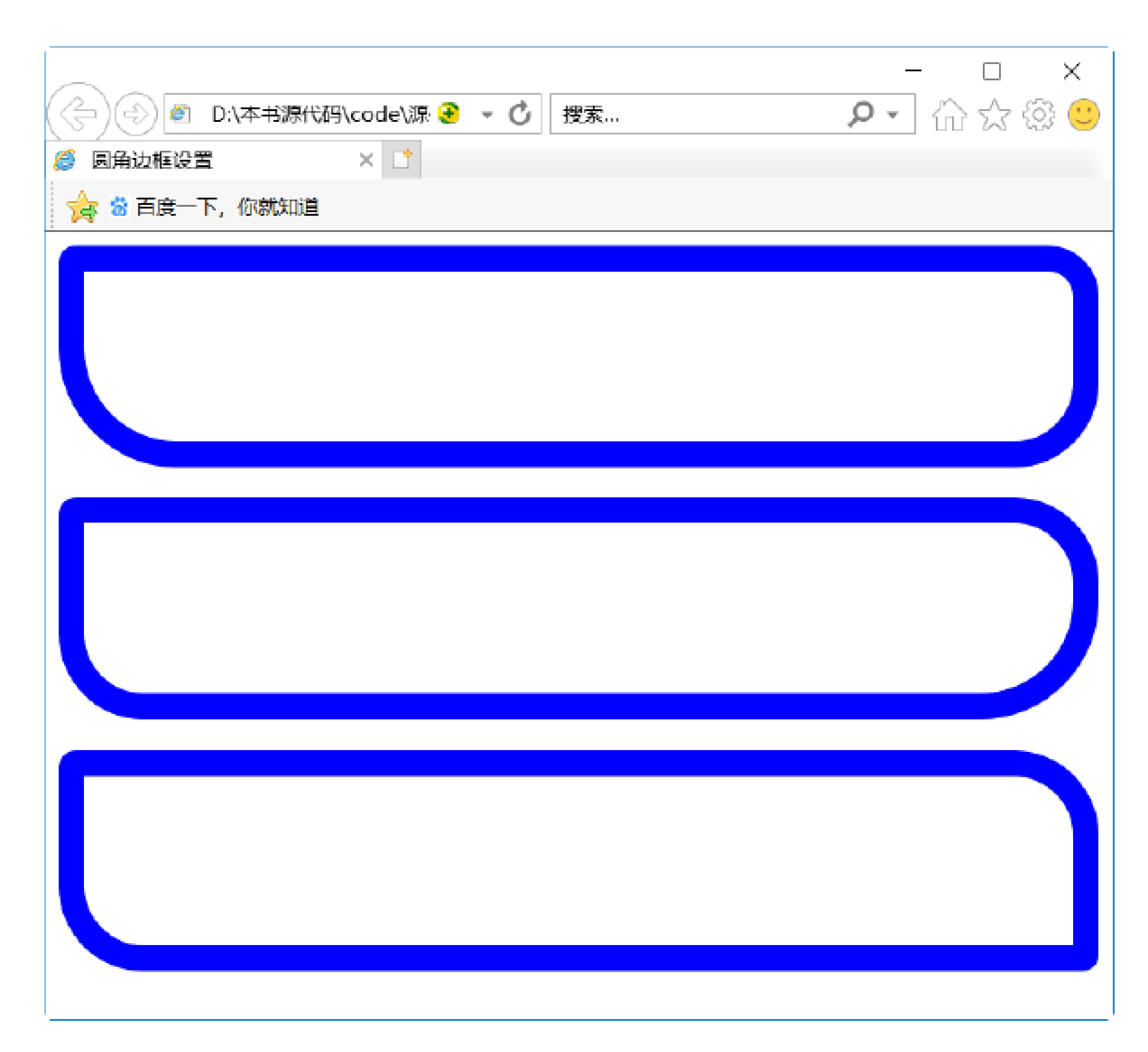


图 13-19　设置 4 个圆角边框

2. border-radius 衍生属性

除了上面设置圆角边框的方法之外，还可以使用如表 13-11 列出的属性单独为相应的边框设置圆角。

表 13-11　border-radius 衍生属性

属性	描述
border-top-right-radius	定义右上角圆角
border-bottom-right-radius	定义右下角圆角
border-bottom-left-radius	定义左下角圆角
border-top-left-radius	定义左上角圆角

【例 13.18】（实例文件：ch13\13.18.html）

```
<!DOCTYPE html>
<html>
<head>
<title>圆角边框设置</title>
<style>
.div{
   border:15px solid blue;
   height:100px;
   border-top-left-radius:70px;
   border-bottom-right-radius:40px;
</style>
</head>
```

```
<body>
<div class=div></div><br />
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-20 所示，可以看到网页中设置的两个圆角边框，分别使用 border-top-left-radius 和 border-bottom-right-radius 指定。

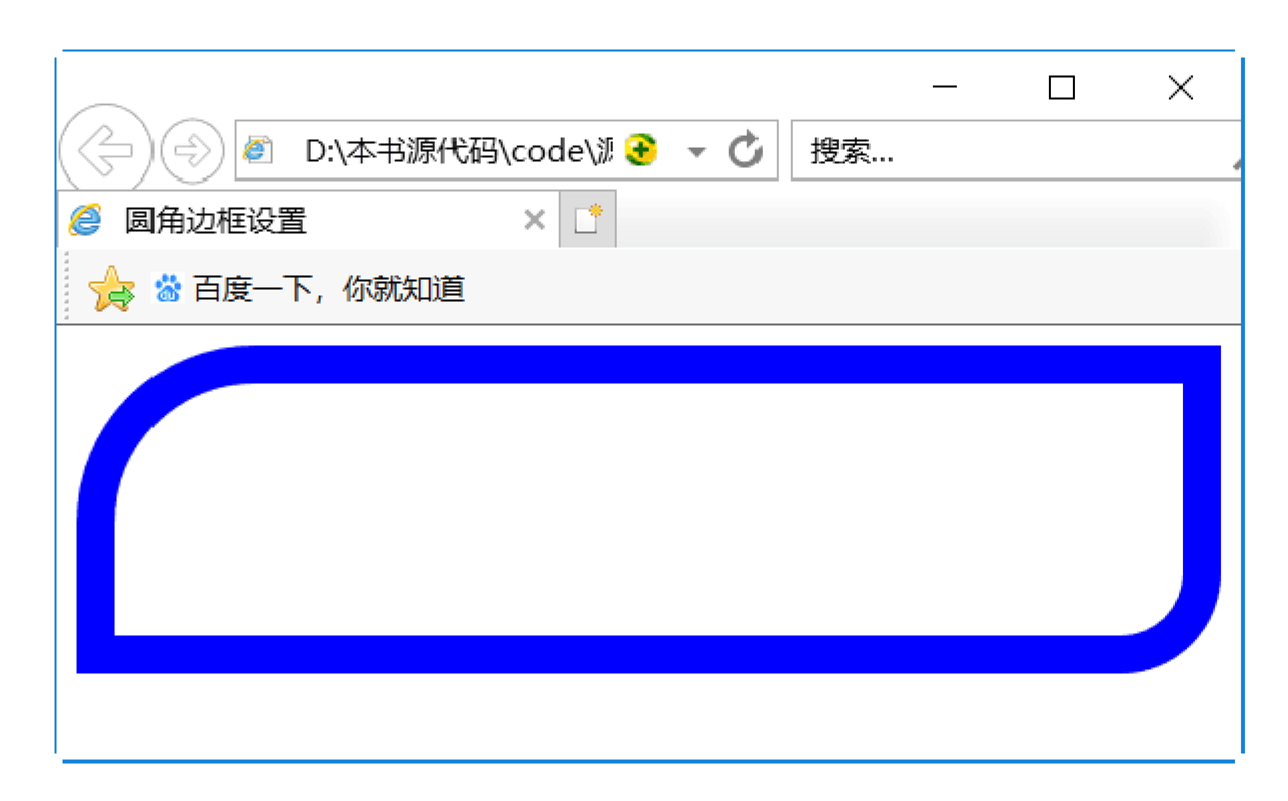


图 13-20 绘制指定圆角边框

13.3.4 绘制边框种类

border-radius 属性可以根据不同的半径值来绘制不同的圆角边框，同样也可以利用 border-radius 来定义边框内部的圆角，即内圆角。需要注意的是，外部圆角边框的半径称为外半径，内边半径等于外边半径减去对应边的宽度，即将边框内部圆的半径称为内半径。

通过外半径和边框宽度的不同设置，可以绘制出不同形状的内边框，如内直角、小内圆角、大内圆角和圆。

【例 13.19】（实例文件：ch13\13.19.html）

```
<!DOCTYPE html>
<html>
<head>
<title>圆角边框设置</title>
<style>
.div1{
   border:70px solid blue;
   height:50px;
   border-radius:40px;}
.div2{
   border:30px solid blue;
   height:50px;
   border-radius:40px;}
.div3{
```

```
    border:10px solid blue;
    height:50px;
    border-radius:60px;}
 .div4{
    border:1px solid blue;
    height:100px;
    width:100px;
    border-radius:50px;}
 </style>
 </head>
 <body>
 <div class=div1></div><br />
 <div class=div2></div><br />
 <div class=div3></div><br />
 <div class=div4></div><br />
 </body>
 </html>
```

在 IE 11.0 中浏览效果如图 13-21 所示，可以看到网页中第一个边框内角为直角，第二个边框内角为小圆角，第 3 个边框内角为大圆角，第 4 个边框为圆。

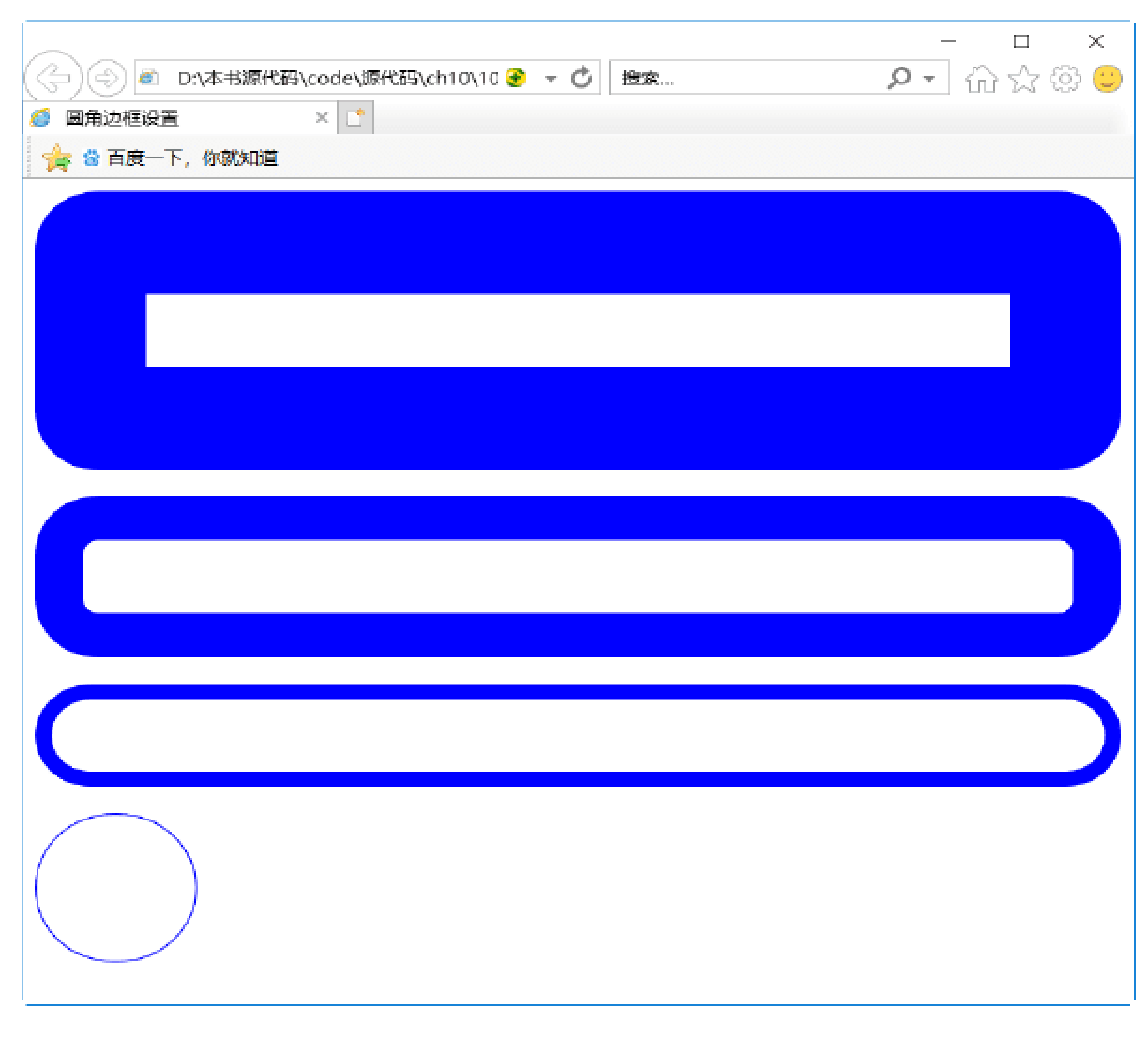


图 13-21　绘制不同种类边框

当边框宽度设置大于圆角外半径，即内半径为 0 时，就会显示内直角，而不是圆直角，所以内外边曲线的圆心必然是一致的（见上例中第一种边框设置）。如果边框宽度小于圆角半径，即内半径小于 0，则会显示小幅圆角效果（见上例中第二个边框设置）。如果边框宽度设置远远小于圆角半径，即内半径远远大于 0，则会显示大幅圆角效果（见上例中第 3 个边框设置）。如果设置元素相同，同时设置圆角半径为元素大小的一半，则会显示圆（见上例中的第 4 个边框设置）。

13.4 图片边框

在 CSS3 中，为了增强边框效果，特意新增了一个属性 border-image，用来控制边框图片。实际上该属性与 background-image 属性功能相似，只不过它的功能更强大一些。

13.4.1 图片边框属性

border-image 是 CSS3 新增核心属性之一，也是一个非常实用的属性，它可以设置 border 边框的背景图。但可惜的是，目前浏览器对其支持不是太好。例如 Webkit 引擎支持-Webkit-border-image 私有属性、Mozilla Gecko 引擎支持-moz0-borer-image 私有属性、Presto 引擎支持-o-border-image 私有属性。

border-image 语法格式如下：

```
    border-image: none|<image> [ <number>|<percentage>]{1,4}
[/<border-width>{1,4}]? [stretch|repeat|round]{0,2}
```

其参数含义如表 13-12 所示。

表 13-12 图片边框属性参数

参数	说明
none	默认值，无背景图
<image>	使用绝对或相对 url 地址指定背景图像
<number>	边框宽度或边框背景图像大小，使用固定像素值表示
<percentage>	设置边框背景图像大小，即边框宽度，用百分比表示
[stretch | repeat | round]	拉伸 | 重复 | 平铺（其中 stretch 是默认值）

为了方便设计师更加灵活地定义边框的背景图像，CSS3 允许将 border-image 属性派生出众多子属性，一方面 CSS3 将 border-image 分成了 8 个部分，使用 8 个子属性分别定义特定方位上边框的背景图像，具体属性如表 13-13 所示。

表 13-13 border-image 派生属性

派生属性	说明
border-top-image	定义顶部边框背景图像
border-right-image	定义右侧边框图像
border-bottom-image	定义底部边框图像
border-left-image	定义左侧边框图像

（续表）

派生属性	说明
border-top-left-image	定义左上角边框图像
border-top-right-image	定义右上角边框图像
border-bottom-left-image	定义左下角边框图像
border-bottom-right-image	定义右下角边框图像

另一方面，border-image 还派生了如表 13-14 所示的几个属性。

表 13-14　border-image 派生属性

派生属性	说明
border-image-source	定义边框的背景图像源，即图像 URL
border-image-slice	定义如何裁切背景图像，与背景图像的定位功能不同
border-image-repeat	定义边框背景图像的重复性
border-image-width	定义边框背景图像的现实大小，虽然 W3C 定义了该属性，但浏览器还是习惯使用 border-width 实现相同的功能
border-image-outset	定义边框背景图像的偏移位置

13.4.2　设置图像边框显示方式

可以看出 border-image 是一个复合属性，包含图像源、剪裁位置和重复性。例如，border-image:url (1.jpg) 50 no-repeat 样式就表示边框背景图像为 1.jpg，剪裁位置 50 像素，禁止重复。border-image 属性使用 url()调用背景图像，图像可以是相对路径或绝对路径，也可以不适用图像，即 border-image:none。

border-image 属性最让人迷惑的地方就是其第二个参数值，剪裁位置，即 boder-image-slice。boder-image-slice 用法比较特殊，该属性值没有单位，默认单位为像素；支持百分比值，百分比值是相对于边框图像而言的。例如，边框图像大小为 400 像素×300 像素，当 border-image-slice 属性值为 20%时，实际的效果就是剪裁了图像的 60 像素、80 像素、60 像素、80 像素的四边大小，然后获取原来图像一定比例的图像。如果 border-image-slice 使用的是数值，就表示按数值大小对原图像进行裁剪。

当使用 boder-image-slice 图像将图片按百分比或数值进行裁剪后，就会获取一定数量的图像，然后将这些图像重新安置到边框背景上，这时图像非常容易发生变形。这个与 CSS 中 clip 属性非常相似。border-image-slice 属性值标准包含 4 个参数，它遵循 CSS 方位规则，按照上、

右、下、左的顺时针方向赋值剪裁。对于第 3 个属性，图像重复性，会在下一个小节介绍。

使用示例如下所示。

```
border-image:url(images/a.jpg) 0 12 0 12 stretch stretch
```

上述代码中，url 表示获取图片；“0 12 0 12”表示按顺时针设置边框宽度，即裁剪图像，也可以简写为“0 12”；第一个“stretch”表示在水平方向上延伸；第二个“stretch”表示在垂直方向上延伸。

【例 13.20】（实例文件：ch13\13.20.html）

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
border:15px solid transparent;
width:300px;
padding:10px 20px;}
#round
{
-moz-border-image:url(1.png) 30 30 round;     /* Old Firefox */
-Webkit-border-image:url(1.png) 30 30 round; /* Safari and Chrome */
-o-border-image:url(1.png) 30 30 round;      /* Opera */
border-image:url(1.png) 30 30 round;}
#stretch
{
-moz-border-image:url(1.png) 30 30 stretch;     /* Old Firefox */
-Webkit-border-image:url(1.png) 30 30 stretch;  /* Safari and Chrome */
-o-border-image:url(1.png) 30 30 stretch;       /* Opera */
border-image:url(1.png) 30 30 stretch;}
</style>
</head>
<body>
<p>使用的原始图片如下：</p>
<img src="1.png">
<p>border-image 属性规定了边框的图片。</p>
<div id="round">图片铺满整个边框。</div>
<br />
<div id="stretch">图片被拉伸以填充该区域。</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-22 所示。

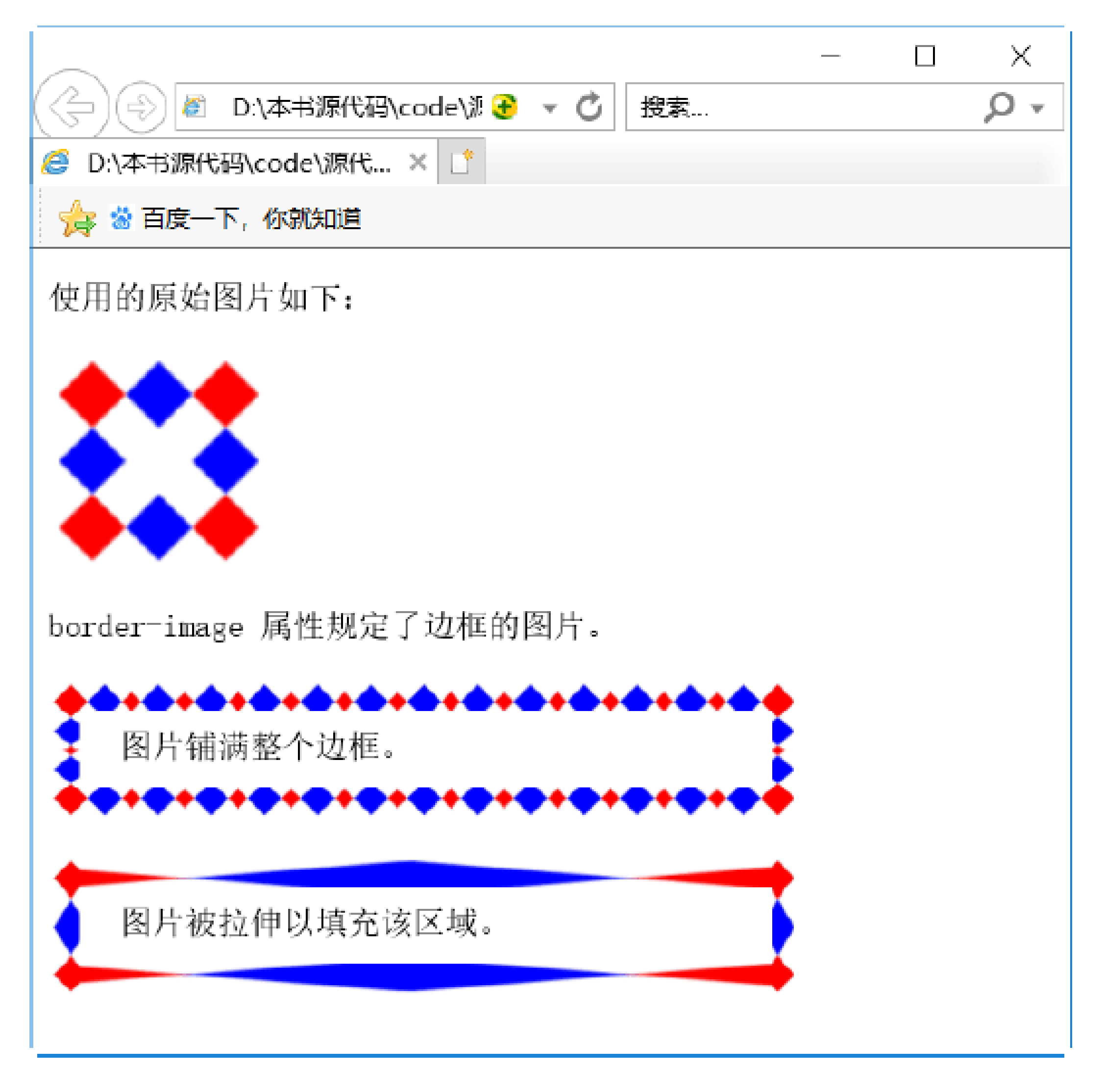


图 13-22　边框背景显示

此时设置下面语句：

```
#round
{
-moz-border-image:url(1.png) 81;     /* Old Firefox */
-Webkit-border-image:url(1.png) 81; /* Safari and Chrome */
-o-border-image:url(1.png) 81;      /* Opera */
border-image:url(1.png) 81;}
```

在 IE 11.0 中浏览效果如图 13-23 所示，可以看到上面图片没有被切割，并分成 4 份在四角显示。

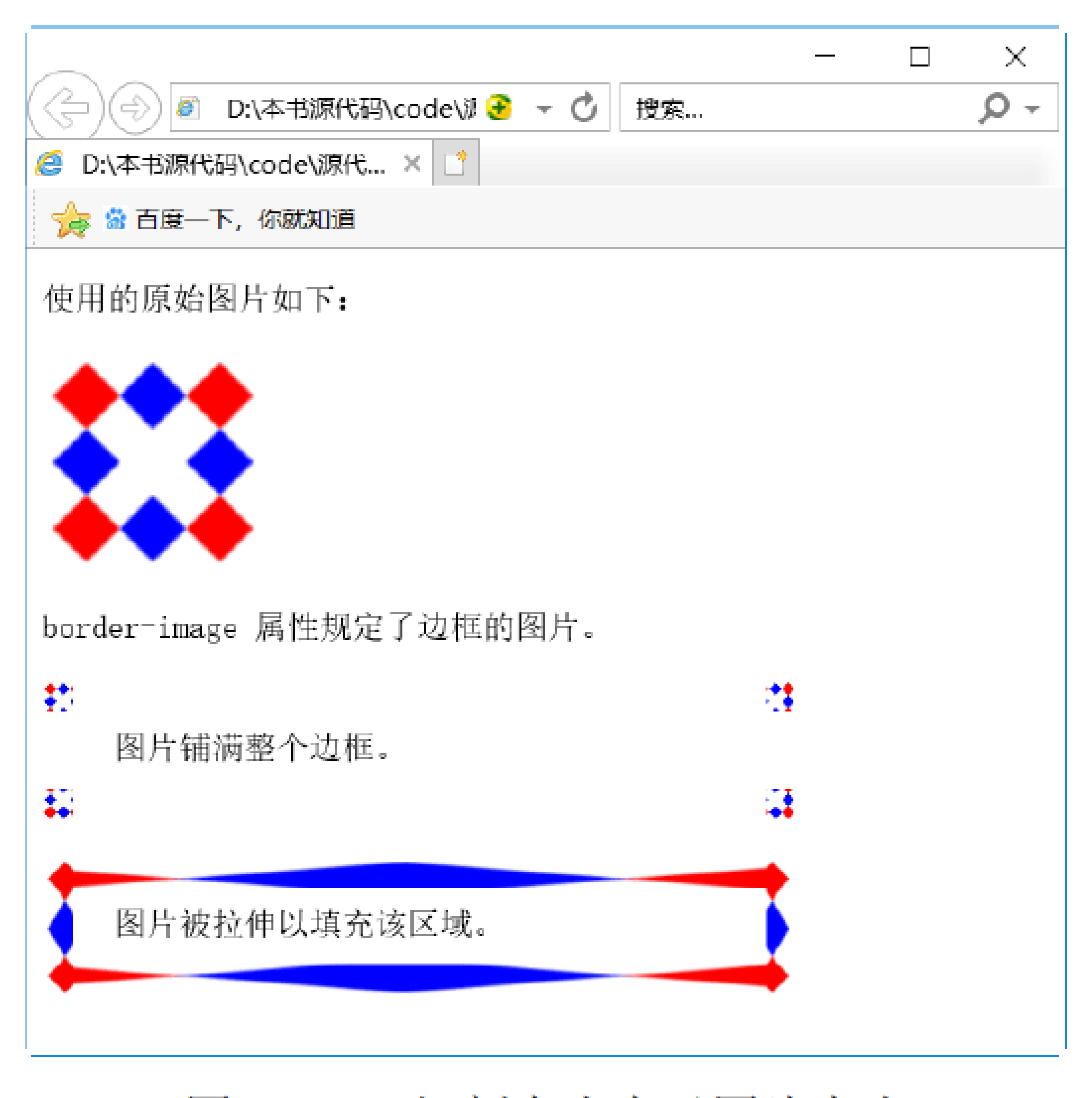


图 13-23　切割大小大于图片大小

13.5 综合实例——设计公司主页

打开各种类型商业网站，最先映入眼帘的就是首页，也称为主页。作为一个网站的门户，主页一般要求版面整洁、美观大方。结合前面学习的背景和边框知识，我们创建一个简单的商业网站。具体步骤如下：

01 分析需求。

在本实例中，主页包括三个部分：一部分是网站 Logo；一部分是导航栏；最后一部分是主页显示内容。网站 logo 此处使用了一个背景图来代替，导航栏使用表格实现，内容列表使用无序列表实现。

02 构建基本 HTML。

为了划分不同的区域，HTML 页面需要包含不同的 div 层，每一层代表一个内容。一个 div 包含背景图；一个 div 包含导航栏；一个 div 包含整体内容，内容又可以划分两个不同的层。

基本 HTML 代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>公司主页</title>
</head>
<body>
<center>
<div>
<div class="div1" align=center></div>
<div class=div2>
<table width=99%><tr align=center><td>首页</td><td>最新消息</td><td>产品展示</td><td>销售网络</td><td>人才招聘</td><td>客户服务</td></tr></table>
</div>
<div class=div3>
<div class=div4>
<ul>最新消息
<li>公司举办2019金秋篮球比赛</li>
<li>消防演练大比武</li>
<li>优秀员工评比</li>
<li>公司发布招聘信息</li>
</ul>
</div>
<div class=div5>
<ul>客户案例
<li>上海电力公司</li>
```

```
<li>浙江电力公司</li>
<li>辽宁电力公司</li>
<li>河北电力公司</li>
</ul>
</div>
</div>
</div>
</center>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 13-24 所示，可以看到在网页中显示了导航栏和两个列表信息。

03 添加 CSS 代码，设置背景 Logo。

```
<style>
.div1{
    height:100px;
    width:820px;
    background-image:url(main.jpg);
    background-repeat:no-repeat;
    background-position:center;
    background-size:cover;}
</style>
```

在 IE 11.0 中浏览效果如图 13-25 所示，可以看到在网页顶部显示了一个背景图，此背景覆盖整个 div 层，并不重复，并且背景图片居中显示。

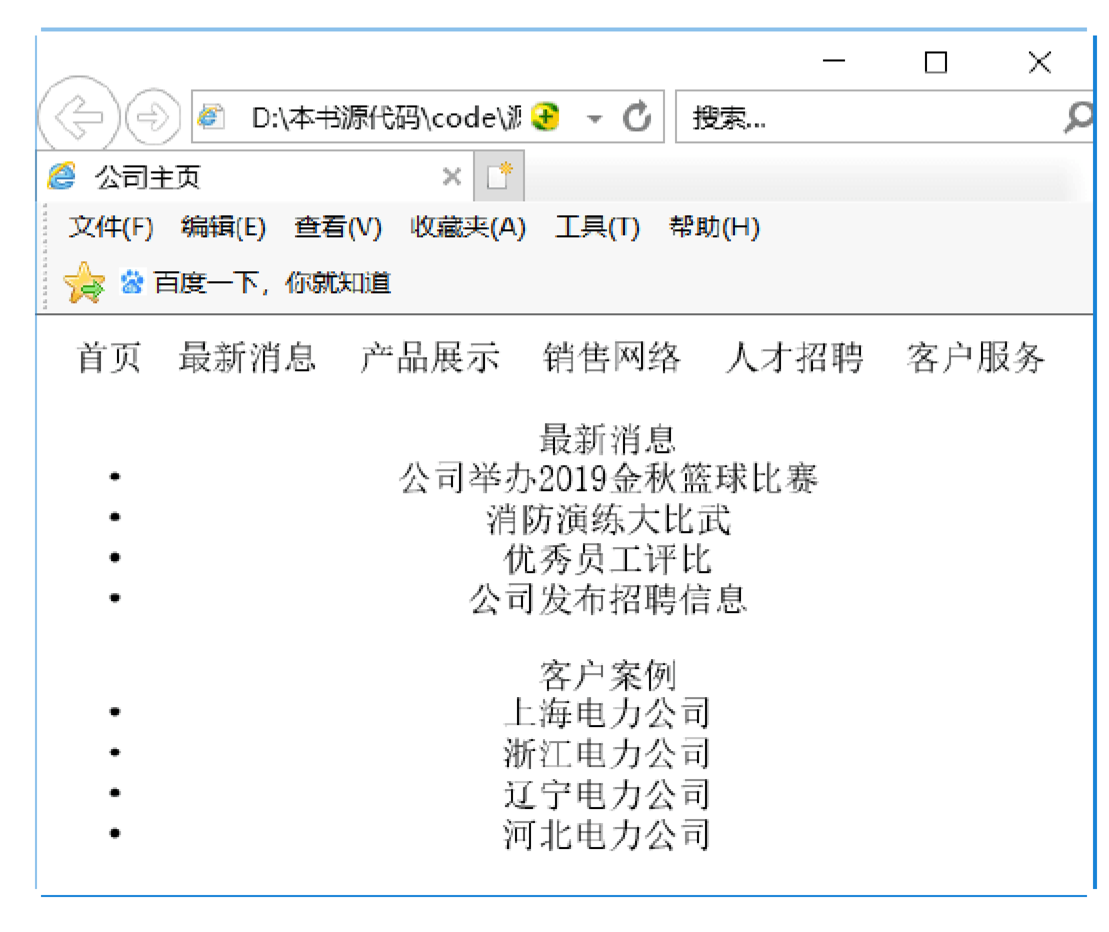


图 13-24 基本 HTML 结构



图 13-25 设置背景图

04 添加 CSS 代码，设置导航栏。

```
.div2{
   width:820px;
   background-color:#d2e7ff;
```

```
}
table{
     font-size:20px;
     font-family:"幼圆";
}
```

在 IE 11.0 中浏览效果如图 13-26 所示，可以看到在网页中导航栏背景色为浅蓝色，表格中字体大小为 20 像素，字体类型是幼圆。

05 添加 CSS 代码，设置内容样式。

```
.div3{
    width:820px;
     height:320px;
     border-style:solid;
     border-color:#ffeedd;
     border-width:10px;
     border-radius:60px;
}
.div4{
    width:810px;
    height:150px;
    text-align:left;
    border-bottom-width: 2px;
    border-bottom-style:dotted;
    border-bottom-color:#ffeedd;
}
.div5{
    width:810px;
    height:150px;
    text-align:left;
}
```

在 IE 11.0 中浏览效果如图 13-27 所示，可以看到在网页中内容显示在一个圆角边框中，两个不同的内容块中间使用虚线隔开。

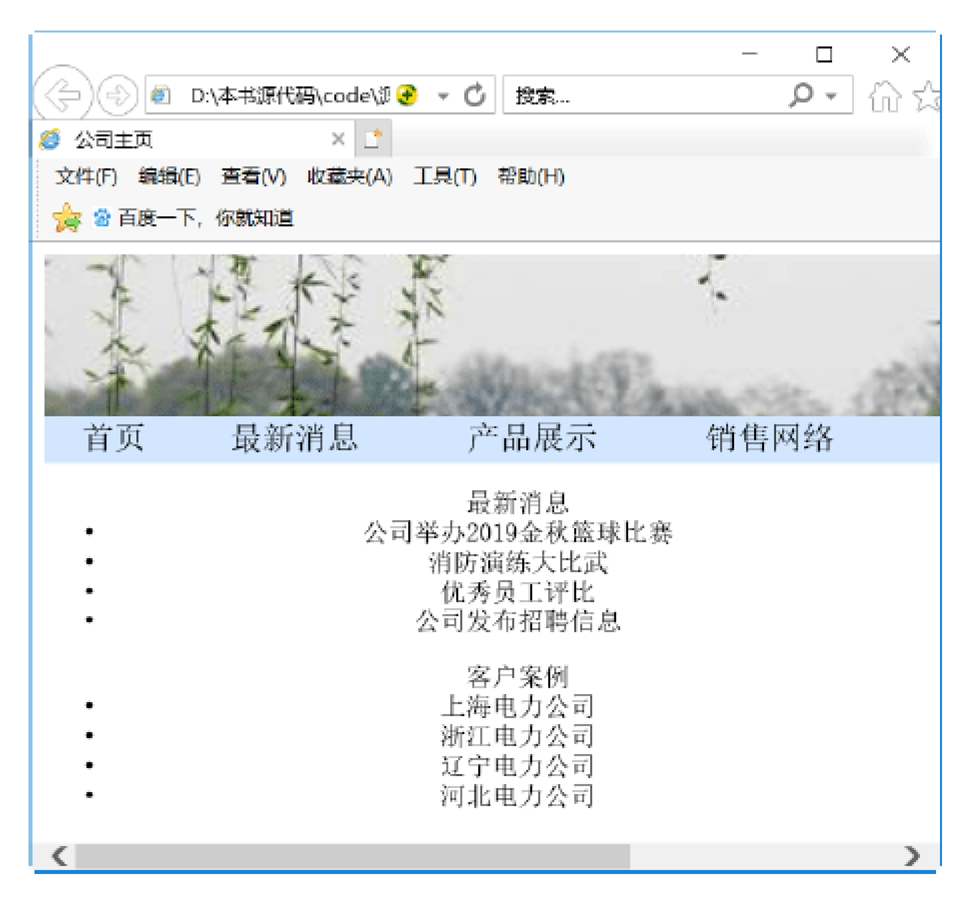


图 13-26 设置导航栏

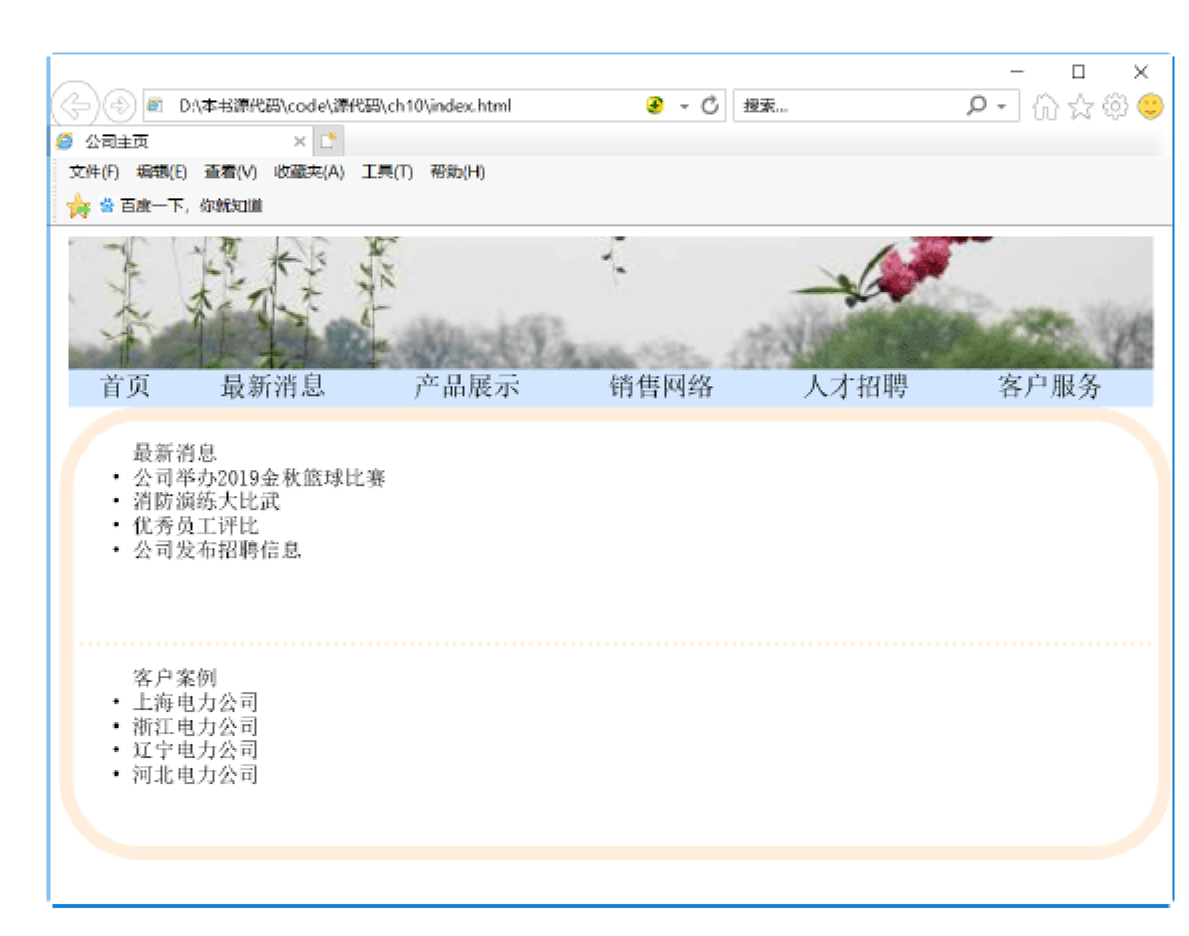


图 13-27 CSS 修饰边框

06 添加 CSS 代码，设置列表样式。

```
ul{

    font-size:20px;
    font-family:"楷体";
}
```

在 IE 11.0 中浏览效果如图 13-28 所示，可以看到在网页中列表字体大小为 20 像素，字体类型为楷体。

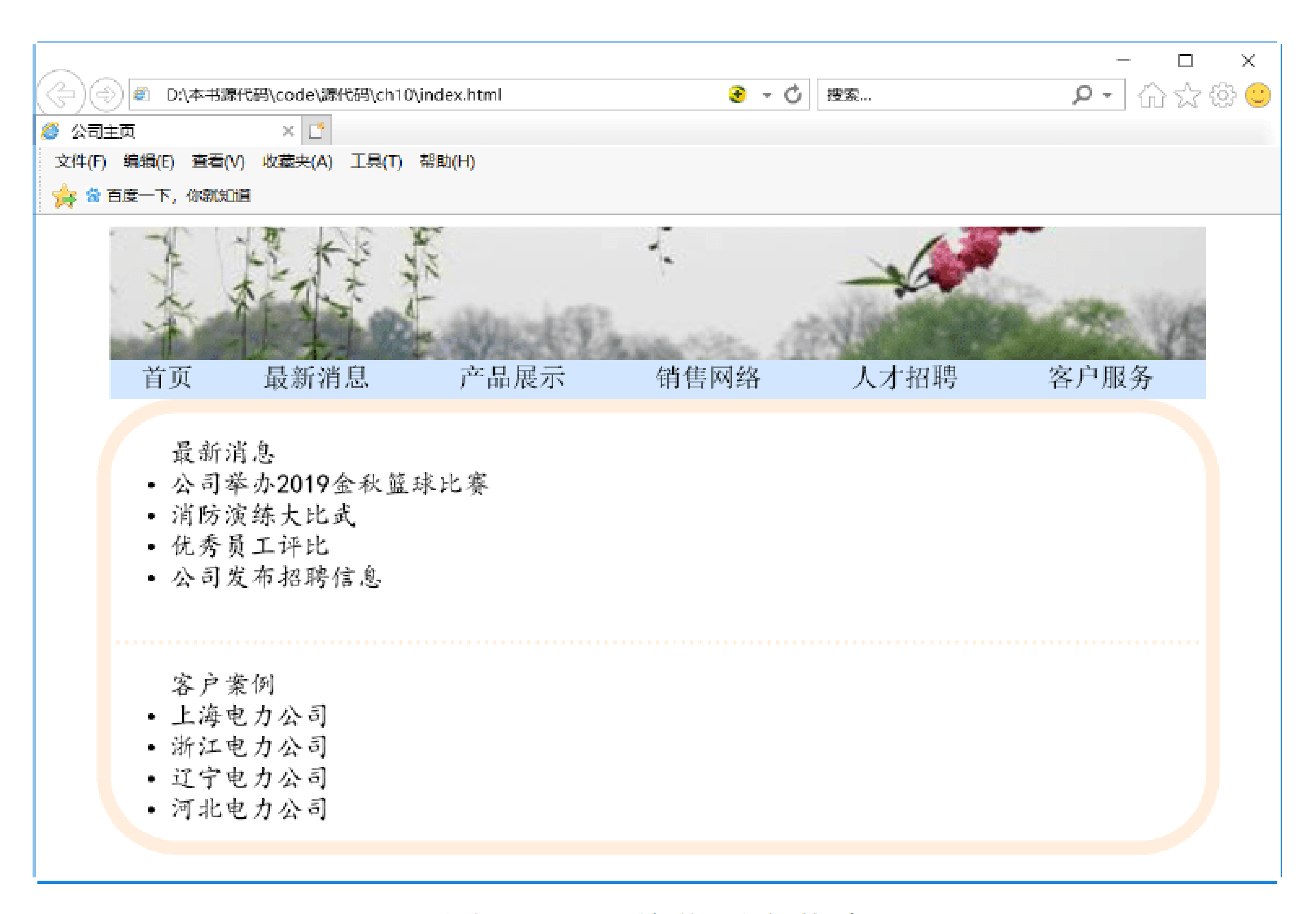


图 13-28　美化列表信息

13.6　专家解惑

1. 我的背景图片不显示，是不是路径的问题呢？

在一般情况下，设置图片路径的代码如下：

```
background-image:url(logo.jpg);
background-image:url(../logo.jpg);
background-image:url(../images/logo.jpg);
```

对于第一种情况“url(logo.jpg)”，要看此图片是不是与 CSS 文件在同一目录。

对于第二和第三种情况，是极力不推荐使用的，因为网页文件可能存在于多级目录中，不同级目录的文件位置注定了相对路径是不一样的。而这样就让问题复杂化了，很可能图片在这个文件中显示正常，换了一级目标，图片就找不到影子了。

有一种方法可以轻松解决这一问题，建立一个公共文件目录，用来存放一些公用图片文件，

如“image”，将图片文件也直接存于该目录中。在 CSS 文件中，代码如下：

```
url(images/logo.jpg)
```

2. 用小图片进行背景平铺好吗？

不要使用过小的图片做背景平铺。这是因为宽高 1 像素的图片平铺出一个宽高 200 像素的区域，需要 200×200=40, 000 次，占用资源。

3. 边框样式 border:0 会占用资源吗？

推荐的写法是 border:none，虽然 border:0 只是定义边框宽度为零，但边框样式、颜色还是会被浏览器解析，占用资源。

第14章　网页的定位与布局

网页设计中，能否很好地定位网页中的每个元素是网页整体布局的关键。一个布局混乱、元素定位不准的页面，是每个浏览者都不喜欢的。而把每个元素都精确定位到合理位置，才是构建美观大方页面的前提。

14.1　定位方式

在 CSS3 中，定位可以将一个元素精确地放在页面上用户所指定的位置，而布局是将整个页面的元素内容整洁且完美地摆放。定位的实现是布局成功的前提。如果清晰地掌握了定位原理，那么就能够创建多种高级而精确的布局，并会让网页更加完美地实现。

14.1.1　定位属性

在网页设计中，定位（Positioning）就是用户精确地定义 HTML 元素框所在页面中的位置，可以是页面绝对位置，也可以处于其上级元素、另一个元素或浏览器窗口的绝对位置。

可以将每个元素都认为包含在一个矩形框内，称为元素框。而元素内容与元素框共同形成了元素块。所谓定位，就是定位元素块位置和大小。实现 CSS3 定位，需要依赖定位属性才能够完成。

表 14-1 列出了 CSS3 中全部有关的定位属性。

表 14-1　定位属性

定位属性	含义
position	定义位置
left	指定元素横向距左部距离
right	指定元素横向距右部距离
top	指定元素纵向距顶部距离
bottom	指定元素纵向距底部距离

（续表）

定位属性	含义
z-index	设置元素的层叠顺序
width	设置元素框宽度
height	设置元素框高度
overflow	内容溢出控制
clip	剪切

表中前 6 个属性是实际的定位属性，后面 4 个相关属性是用来设置元素框，或对元素框中的内容进行控制的属性。其中，position 属性是最主要的定位属性，它既可以定义元素框的绝对位置，又可以定义相对位置，而 left、right、top 和 bottom 只在 position 属性中使用才会发挥作用。

14.1.2 position 定位

网页中各种元素需要有自己合理的，从而搭建整个页面的结构。在 CSS3 中，可以通过 position 这个属性，对页面中元素进行定位。

语法格式如下：

```
position : static | absolute | fixed | relative
```

其参数含义如表 14-2 所示。

表 14-2 position 属性参数值

参数名	说明
static	元素定位的默认值，无特殊定位，对象遵循 HTML 定位规则，不能通过 z-index 进行层次分级
relative	相对定位，对象不可重叠，可以通过 left、right、bottom 和 top 等属性在正常文档中偏移位置，可以通过 z-index 进行层次分级
absolute	生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。元素的位置通过 “left” “top” “right”及“bottom”属性进行规定
fixed	fixed 生成绝对定位的元素，相对于浏览器窗口进行定位。元素的位置通过“left”“top”“right”及“bottom”属性进行规定

1. 绝对定位 absolute

绝对定位是参照浏览器的左上角，配合 top、left、bottom 和 right 进行定位的，如果没有设置上述的 4 个值，则默认依据父级的坐标原点为原始点的。绝对定位可以通过上、下、左、右来设置元素，使之处在任何一个位置。

在父层 position 属性为默认值时，上、下、左、右的坐标原点以 body 的坐标原点为起始位置。绝对定位的语法格式如下：

```
position:absolute
```

只要将上面代码加入到样式中，使用样式的元素就可以以绝对定位的方式显示了。

【例 14.1】（实例文件：ch14\14.1.html）

```
<!DOCTYPE html>
<html>
<head>
<title>定位属性</title>
</head>
<body>
<div style="background-color: Black; width:200px; height:200px">
<h2 style=" position:absolute; left:80px; top:80px; width:110px;
height:50px;background-color:Red;">这是绝对定位</h2>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-1 所示，可以看到红色元素框依据浏览器左上角为原点，坐标位置为（80 像素，80 像素），宽度为 110 像素，高度为 50 像素。

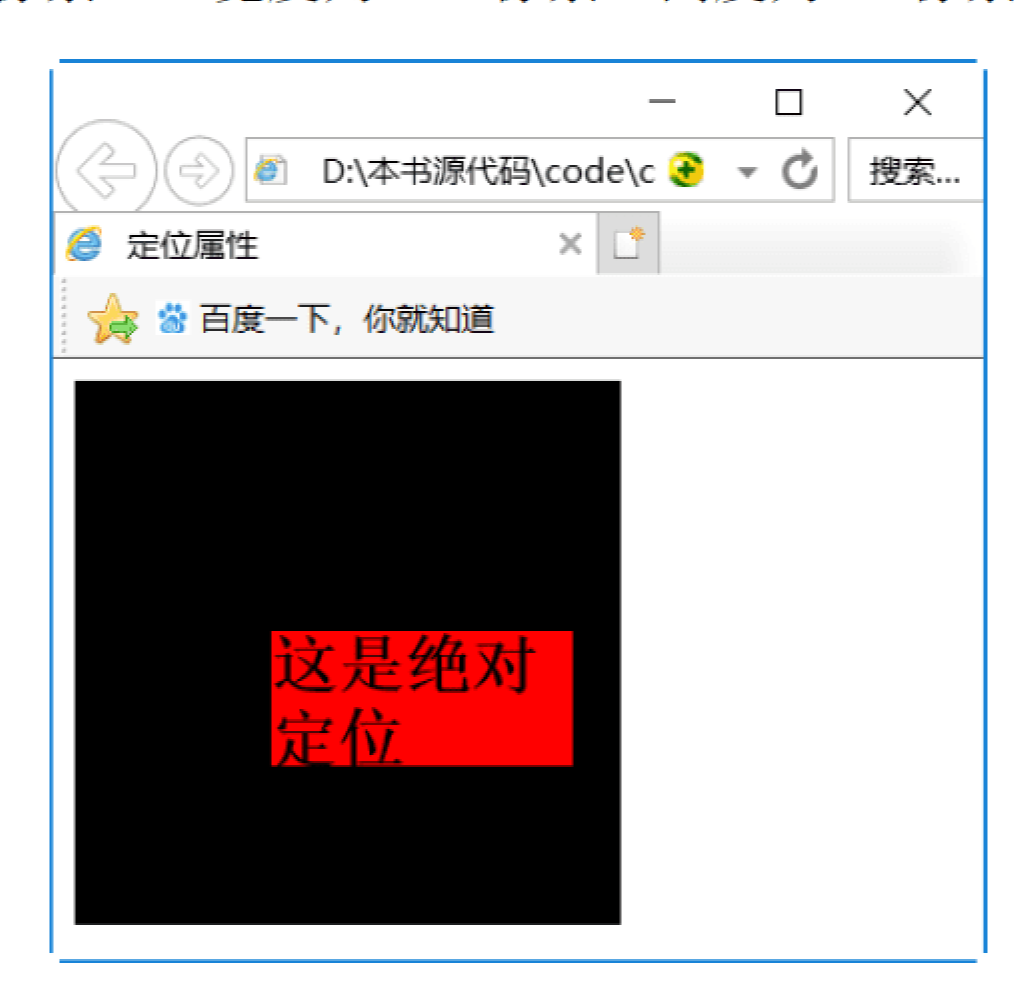


图 14-1　绝对定位



使用绝对定位会产生一个问题。目前，大多数的网页都是居中显示的，而且元素与元素之间布局是紧密的。而绝对定位的开始位置是浏览器左上角的 0 点，当设置各元素块边偏移属性时，由于客户端屏幕分辨率的不同，各元素块的显示可能会有偏差。这是由于页面的显示是随着分辨率的大小而自动适应，而各元素块在参照绝对定位的位置显示，那么在浏览器的视野范围内原始页面可以或超出或缩小的显示。

优秀的页面设计是能够适用各种屏幕分辨率，并且能够保证正常显示的。要解决这个问题，在定位时最好使用相对定位。

2. 相对定位 relative

如果对一个元素进行相对定位，首先它将出现在它所在的位置上，然后通过设置垂直或水平位置，让这个元素“相对于”它的原始起点进行移动。再一点，相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其他框。

绝对定位与相对定位的区别在于：绝对定位的坐标原点为上级元素的原点，与上级元素有关；相对定位的坐标原点为本身偏移前的原点，与上级元素无关。

相对定位的语法格式如下：

```
position:relative
```

【例 14.2】（实例文件：ch14\14.2.html）

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
h2.pos left
{
position:relative;
left:-20px
}
h2.pos right
{
position:relative;
left:20px
}
</style>
</head>
<body>
<h2>这是位于正常位置的标题</h2>
<h2 class="pos left">这个标题相对于其正常位置向左移动</h2>
<h2 class="pos right">这个标题相对于其正常位置向右移动</h2>
<p>相对定位会按照元素的原始位置对该元素进行移动。</p>
<p>样式 "left:-20px" 从元素的原始左侧位置减去 20 像素。</p>
<p>样式 "left:20px" 向元素的原始左侧位置增加 20 像素。</p>
```

```
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-2 所示，可以看到页面显示了 3 个标题，最上面标题正常显示，下面两个标题分别以正长标题为原点，向左或向右分别移动了 20 像素。

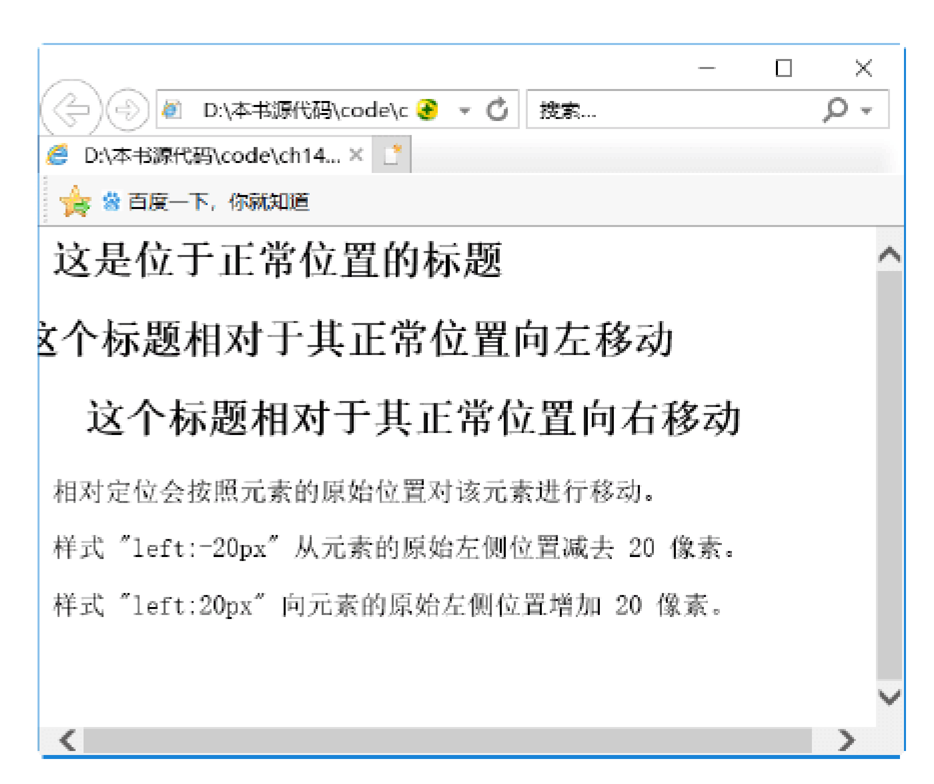


图 14-2　相对定位

3. 固定定位 fixed

固定定位和绝对定位比较相似，它是绝对定位的一种特殊形式，固定定位的容器不会随着滚动条的拖动而变化位置。在视线中，固定定位的容器位置是不会改变的。固定定位可以把一些特殊效果固定在浏览器的视线位置。

由于固定定位的参照位置不是上级元素块而是浏览器窗口，所以可以使用固定定位来设置类似传统框架样式布局，以及广告框架或导航框架等。使用固定定位的元素可以脱离页面，无论页面如何滚动，始终处在页面的同一位置上。

固定定位语法格式如下：

```
position:fixed
```

【例 14.3】（实例文件：ch14\14.3.html）

```
<!DOCTYPE html>
<html>
<head>
<title>CSS固定定位</title>
<style type="text/css">...
*{
padding:0;
margin:0;
}
#fixedLayer {
width:100px;
line-height:50px;
```

```
background: #FC6;
border:1px solid #F90;
position:fixed;
left:10px;
top:10px;
}
</style>
</head>
<body>
<div id="fixedLayer">固定不动</div>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-3 所示，可以看到拖动滚动条时，无论页面内容怎么变化，其黄色框“固定不动”始终处在页面左上角顶部。

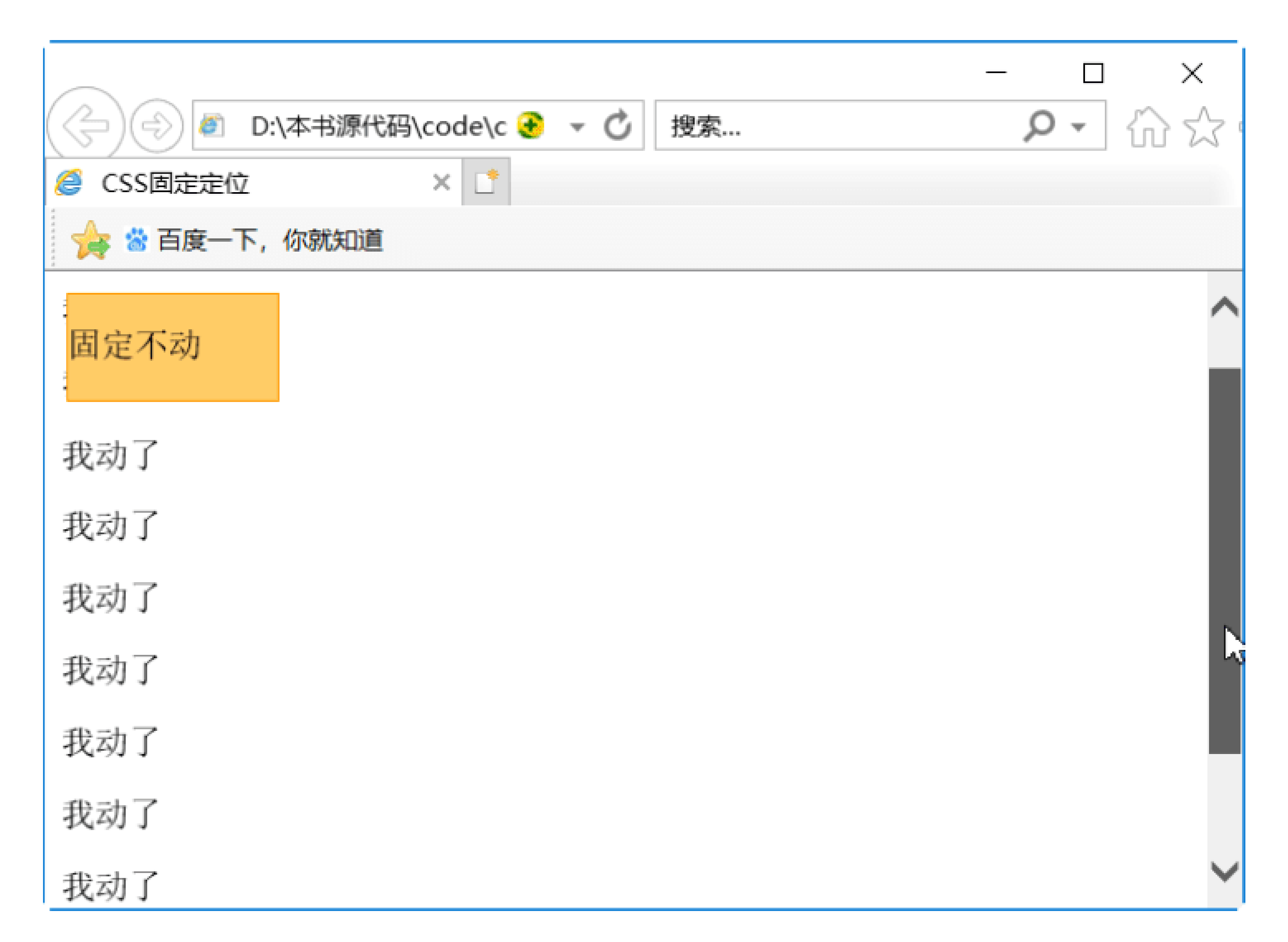


图 14-3 固定定位

14.1.3 层叠顺序 z-index

对 HTML 元素进行定位时，可以从其高度、宽度和深度 3 个方面入手，高度使用 height、宽度使用 width、深度使用 z-index。z-index 用来设置元素层叠的次序，其方法是每个元素指定一个数字，数字较大的元素将叠加在数字较小的元素之上。

z-index 语法格式如下：

```
z-index : auto | number
```

其参数值 auto 表示遵循父对象的定位，number 是一个无单位的整数值，可以为负值。如果两个决定定位元素的 z-index 属性具有相同的 number 值，则依据该元素在 HTML 文档中声明的顺序进行层叠。如果绝对定位的元素没有指定 z-index 属性，则此属性的 number 值为正数的元素会叠加在该元素之上，而 number 值为负数的对象在该元素之下。如果将参数设置为 null，则可以消除此属性。该属性只作用于 position 的属性值为 relative 或 absolute 的对象，不能作用在窗口组件上。

【例 14.4】（实例文件：ch14\14.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>Z-index使用</title>
<style>
#big {
width:800px;
height:220px;
padding:6px;
background-color:#999999;
position:relative;
}
#Div1 {
width:160px;
height:80px;
background-color:#FFD700;
padding:6px;
position:absolute;
left:9px;
top:9px;
z-index:6;
}
#Div2 {
width:120px;
height:80px;
background-color:thistle;
```

```
    padding:6px;
    position:absolute;
    left:280px;
    top:90px;
    z-index:4;
    }
    #Div3 {
    width:140px;
    height:80px;
    background-color:lightskyblue;
    padding:6px;
    position:absolute;
    left:150px;
    top:25px;
    z-index:5;
    }
    </style>
    </head>
    <body>
    <div id="big">
    <div id="Div1"><br />
      z-index值是6 ; </div>
    <div id="Div2"><br />
      z-index值是4 ; </div>
    <div id="Div3"><br />
      z-index值是 5 ; </div>
    </div>
    </body>
    </html>
```

在 IE 11.0 中浏览效果如图 14-4 所示，可以看到网页中显示了 3 个层，3 个层中数值大小不同，并从大到小分别在别的层上显示。

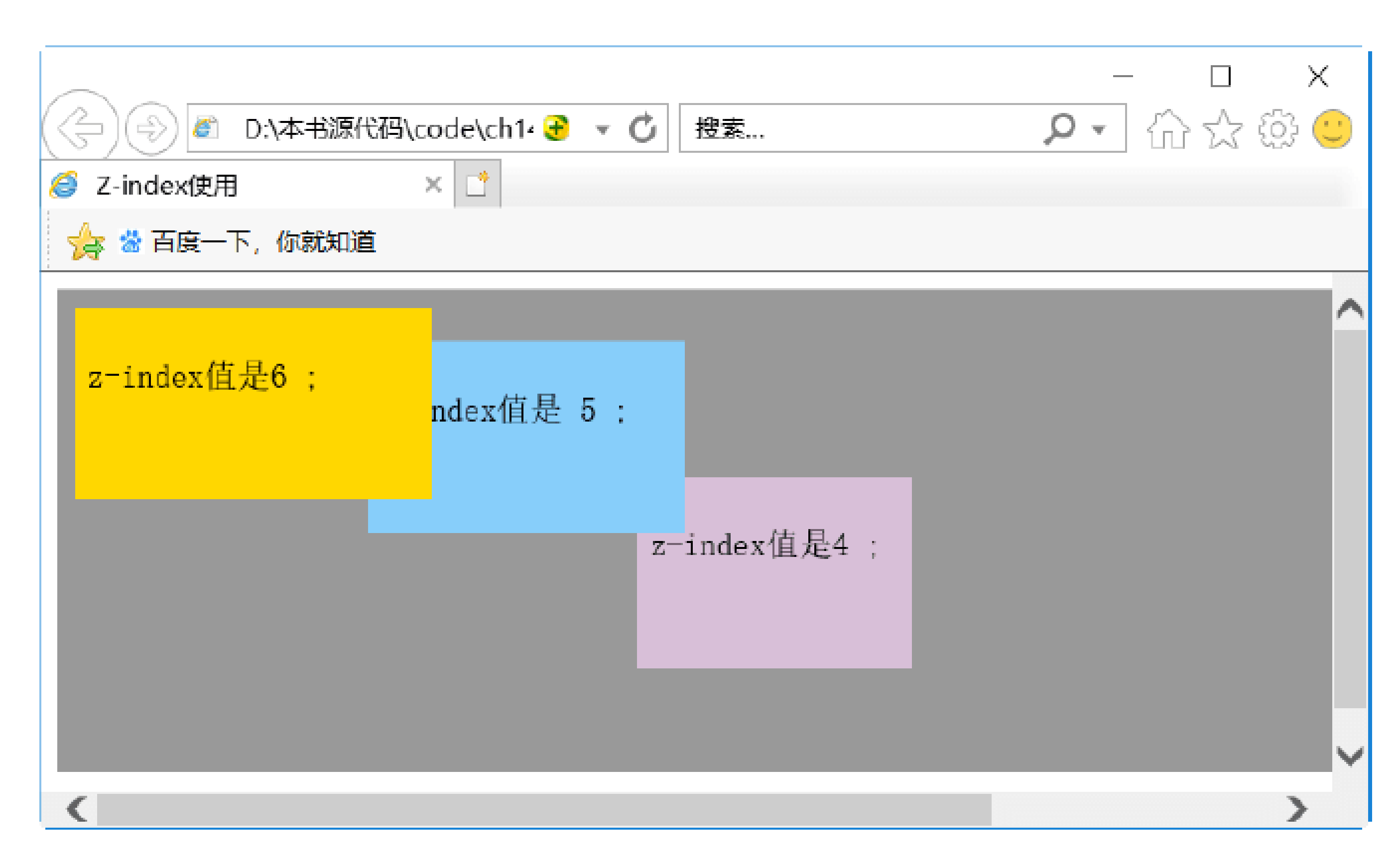


图 14-4　z-index 效果显示

14.1.4 边偏移属性

边偏移属性包含 left、right、top 和 bottom。所谓边偏移属性，就是用来描述元素块与包含元素块最近的边线之间的偏移量的属性。其中 left 描述元素块最左边与包含其边框最左边的边线的距离，如果 left 属性值为正，则会偏右移；如果为负；则会使它更向左移，甚至移出边线，其他依次类推。

left、right、top 和 bottom4 个属性取值非常相似，这里以 left 为例进行介绍。left 语法格式如下：

```
left : auto | length
```

上面参数值中 auto 表示系统自动取值；length 表示由浮点数字和单位标识符组成的长度值或百分数。直接设置数值用来设置元素的绝对位置，一旦该位置确定，该元素将始终处于页面中的该位置。使用百分比设置元素位置，是相对于其上级元素的位置而设置的。如果取值为 auto，则在定位中允许元素刚好有显示其内容所需的宽度及高度，而不必再指明宽度及高度的值了。

【例 14.5】（实例文件：ch14\14.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>定位属性</title>
</head>
<body>
<div style="background-color: Black; width:200px; height:200px">
  <p style=" position:relative; left:50%; right:0; top:50%; bottom:0;
width:100px; height:100px;
  background-color:Red;">边偏移</p>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-5 所示，可以看到黄色正方形框在指定位置显示，其底侧和右侧分别与大的矩形框对应。

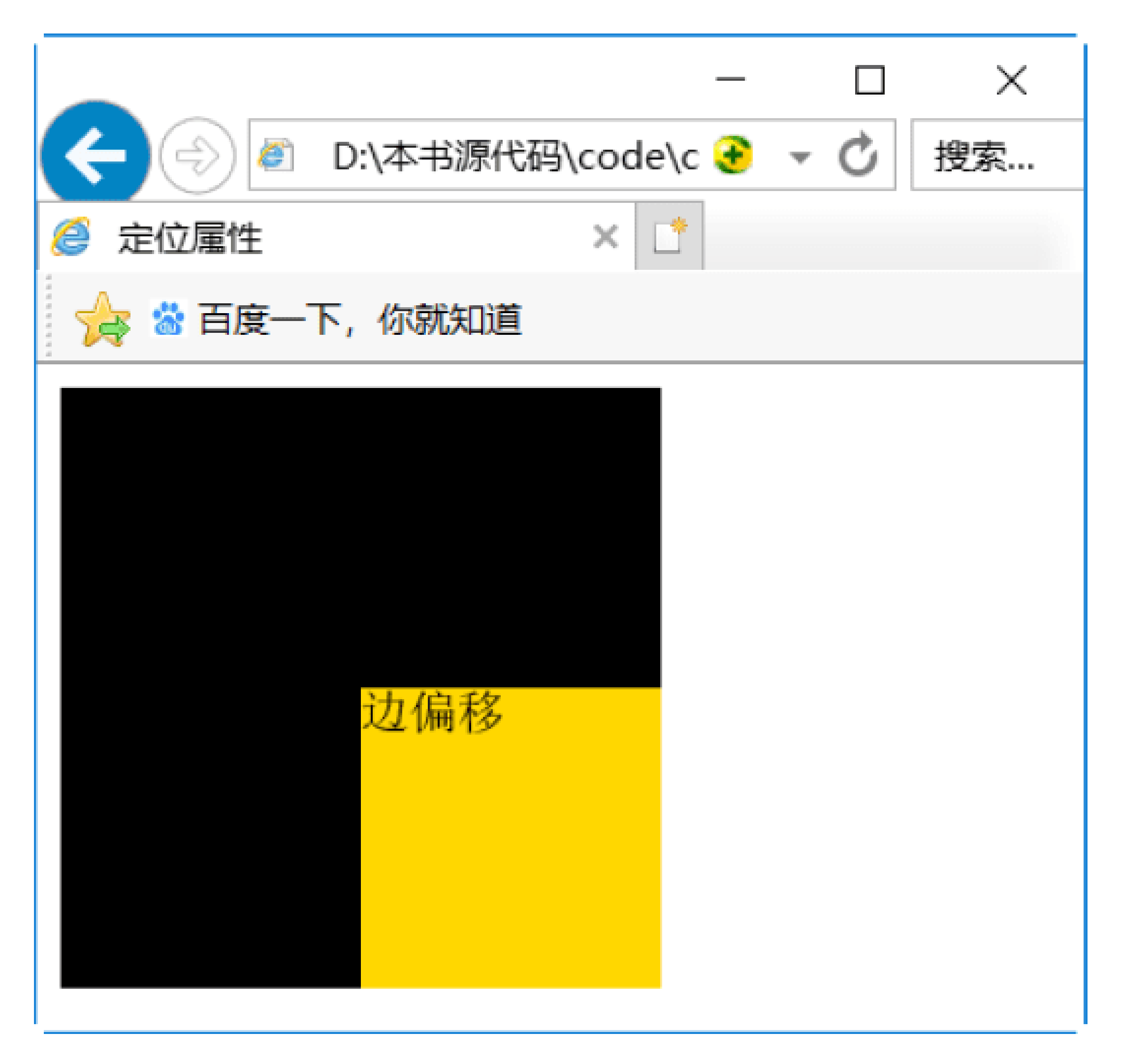


图 14-5　边偏移效果显示

14.2　float 浮动定位

除了使用 position 进行定位外，还可以使用 float 定位。float 定位只能在水平方向上定位，而不能在垂直方式上定位。float 属性表示浮动属性，它用来改变元素块的显示方式。

float 语法格式如下：

```
float : none | left |right
```

其属性值如表 14-3 所示。

表 14-3　float 属性值

属性值	说明
none	元素不浮动
left	浮动在左面
right	浮动在右面

实际上，使用 float 可以实现两列布局，也就是让一个元素在左浮动，一个元素在右浮动，并控制好这两个元素的宽带。

【例 14.6】（实例文件：ch14\14.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>float定位</title>
```

```
<style>
* {
padding:0px;
margin:0px;
}
.big {
width:600px;
height:100px;
margin:0 auto 0 auto;
border:#332533 1px solid;
}
.one {
width:300px;
height:20px;
float:left;
border:#996600 1px solid;
}
.two {
width:290px;
height:20px;
float:right;
margin-left:5px;
display:inline;
border:#FF3300 1px solid;
}
</style>
</head>
<body>
<div class="big">
  <DIV class="one">
  <p>非诚勿扰</p>
  </DIV>
  <DIV class="two">
  <p>中国达人秀</p>
  </DIV>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-6 所示，可以看到显示了一个大矩形框，大矩形框中存在两个小的矩形框，并且并列显示。

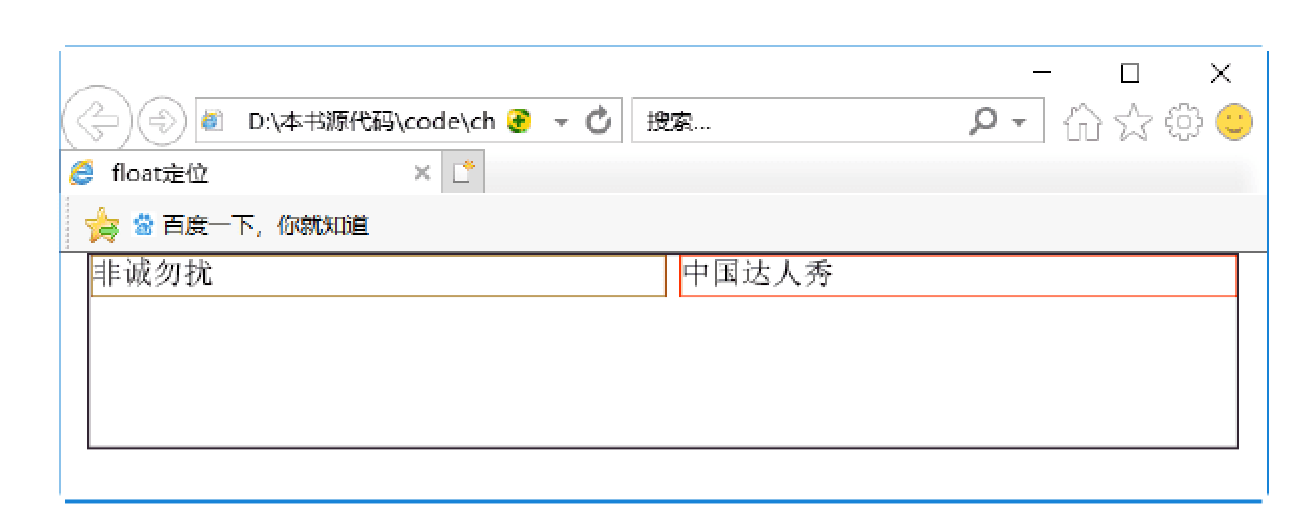


图 14-6 float 浮动布局

使用 float 属性不但改变元素的显示位置，而且会对相邻内容造成影响。定义了 float 属性的元素会覆盖在其他元素上，而被覆盖的区域将处于不可见状态。使用该属性能够实现内容环绕图片的效果。

如果不想让 float 下面的其他元素浮动环绕在该元素周围，就可以使用 CSS3 中的 clear 属性，清除这些浮动元素。

clear 语法格式如下：

```
clear : none | left |right | both
```

其中，none 表示允许两边都可以有浮动对象；both 表示不允许有浮动对象；left 表示不允许左边有浮动对象；right 表示不允许右边有浮动对象。使用 float 以后，在必要的时候就需要通过 clear 语句清除 float 带来的影响，以免出现“其他 DIV 跟着浮动”的效果。

14.3 overflow 溢出定位

如果元素框被指定了大小，而元素的内容不适合该大小，例如元素内容较多，元素框显示不小，此时则可以使用溢出属性 overflow 来控制这种情况。

overflow 语法格式如下：

```
overflow : visible | auto | hidden | scroll
```

各属性值及其说明如表 14-4 所示。

表 14-4 overflow 属性值

属性值	说明
visible	若内容溢出，则溢出内容可见
hidden	若内容溢出，则溢出内容隐藏
scroll	保持元素框大小，在框内应用滚动条显示内容
auto	等同于 scroll，它表示在需要时应用滚动条

overflow 属性适用于以下情况。

（1）当元素有负边界时。

（2）元素框宽于上级元素内容区，换行不被允许。

（3）元素框宽于上级元素区域宽度。

（4）元素框高于上级元素区域高度。

（5）元素定义了绝对定位。

【例 14.7】（实例文件：ch14\14.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title>overflow属性</title>
<style >
div{
position:absolute;
color:#445633;
height:200px;
width: 30%;
float:left;
margin: 0px;
padding: 0px;
border-right: 2px dotted #cccccc;
border-bottom: 2px solid #cccccc;
padding-right: 10px;
overflow:auto;
}
</style>
</head>
<body >
<div>
<p>综艺节目排名</p><p>1 非诚勿扰</p><p>2 康熙来</p>
<p>3  快乐大本营</p><p>4  娱乐大风暴</p><p>5天天向上</p><p>6 爱情连连看</p>
<p>7 锵锵三人行</p><p>8 我们约会吧</p>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-7 所示，可以看到在一个元素框中显示了多个元素，拖动显示的滚动条可以查看全部元素。如果 overflow 设置的值为 hidden，则会隐藏多余元素。

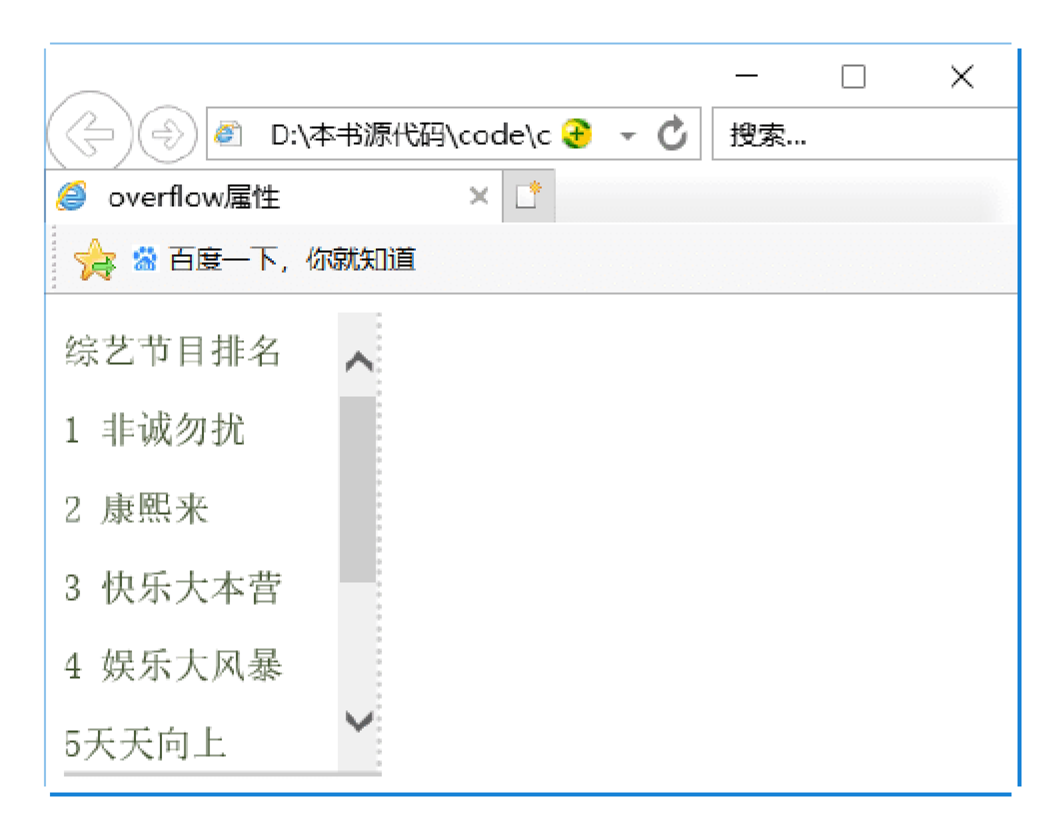


图 14-7 溢出定位

14.4 visibility 隐藏定位

visibility 属性指定是否显示一个元素生成的元素框。这意味着元素仍占据其本来的空间，不过可以完全不可见，即设置元素的可见性。

visibility 语法格式如下：

```
visibility : inherit | visible | collapse | hidden
```

其属性值如表 14-5 所示。

表 14-5 visibility 属性值

属性值	说明
visible	元素可见
hidden	元素隐藏
collapse	主要用来隐藏表格的行或列。隐藏的行或列能够被其他内容使用。对于表格外的其他对象，其作用等同于 hidden。

如果元素 visibility 属性的属性值设置为 hidden，则表现为元素隐藏，即不可见。但是，元素不可见，并不等同于元素不存在，它仍旧会占有部分页面位置，影响页面的布局，就如同可见一样。换句话说，元素仍然处于页面中的位置上，只是无法看到它而已。

【例 14.8】（实例文件：ch14\14.8.html）

```
<!DOCTYPE html>
<html>
<head>
<title>float属性</title>
<style type="text/css">
```

```
.div{
    padding:5px;
}
.pic{
    float:left;
    padding:20px;
    visibility:visible;
}
h1{
   font-weight:bold;
   text-align:center
}
</style>
</head>
<body>
<h1>插花</h1>
<div class="div">
<div class="pic">
  <img src="08.jpg"  width=150px height=100px /></div>
  <p>插花就是把花插在瓶、盘、盆等容器里，而不是栽在这些容器中。所插的花材，枝、花、叶均不带根，只是植物体上的一部分，并且不是随便乱插的，而是根据一定的构思来选材，遵循一定的创作法则，插成一个优美的形体（造型），借此表达一种主题，传递一种感情和情趣，使人赏心悦目，获得精神上的美感和愉快。</p>
<p>在我国插花的历史源远流长，发展至今已为人们日常生活所不可缺少。一件成功的插花作品，并不一定要选用名贵的花材、高价的花器。一般看来并不起眼的绿叶、一个花蕾，甚至路边的野花野草，常见的水果、蔬菜等，都能插出一件令人赏心悦目的优秀作品来。使观赏者在心灵上产生共鸣是创作者唯一的目的，如果不能产生共鸣，那么这件作品就失去了观赏价值。具体地说，插花作品在视觉上首先要立即引起一种感观和情感上的自然反应，如果未能立刻产生反应，那么摆在眼前的这些花材将无法吸引观者的目光。在插花作品中引起观赏者情感产生反应的要素有三点：一是创意（立意），指的是表达什么主题，应选什么花材；二是构思（构图），指的是这些花材怎样巧妙配置造型，在作品中充分展现出各自的美；三是插器，指的是与创意相配合的插花器皿。三者有机配合，作品便会给人以美的享受。</p>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-8 所示。可以看到图片在左边显示，并被文本信息所环绕。此时 visibility 属性为 visible，表示图片可以看见。



图 14-8 隐藏定位显示

14.5 块和行内元素 display

在网页设计中，根据需要可以将段落设置成一个块显示，并带有边框，即类似于 div 层的效果，也可以把多个 HTML 元素放在同一行显示。本节将介绍这两种实现方式。

14.5.1 块元素

在 CSS3 中可以通过 display 属性控制元素显示，即元素显示方式。

display 语法格式如下：

```
    display : block | none | inline | compact | marker | inline-table | list-item
| run-in | table |table-caption | table-cell | table-column | table-column-group
| table-footer-group | table-header-group | table-row | table-row-group
```

其属性值如表 14-6 所示。

表 14-6 display 属性

属性值	说明
block	以块元素方式显示
inline	以内联元素方式显示
none	元素隐藏
list-item	以列表方式显示
compact	CSS2 分配对象为块对象或基于内容之上的内联对象
marker	CSS2 指定内容在容器对象之前或之后。要使用此参数，对象必须与:after 及:before 伪元素一起使用
inline-table	CSS2 将表格显示为无前后换行的内联对象或内联容器

（续表）

属性值	说明
list-item	CSS1 将块对象指定为列表项目，并可以添加可选项目标志
run-in	CSS2 分配对象为块对象或基于内容之上的内联对象
table	CSS2 将对象作为块元素级的表格显示
table-caption	CSS2 将对象作为表格标题显示
table-cell	CSS2 将对象作为表格单元格显示
table-column	CSS2 将对象作为表格列显示
table-column-group	CSS2 将对象作为表格列组显示
table-header-group	CSS2 将对象作为表格标题组显示
table-footer-group	CSS2 将对象作为表格脚注组显示
table-row	CSS2 将对象作为表格行显示
table-row-group	CSS2 将对象作为表格行组显示

display 属性的默认值为 block，即元素的默认显示方式是以块元素方式显示。常用的段落 p、标题 h1、表单 form、列表 ul 和列表选项 li 都可以定义成块元素。一个块元素，其行高、顶部和底部都是可控制的。如果不设置宽度，块就会默认为整个容器的 100%；如果设置了值，显示大小就由值决定。

【例 14.9】（实例文件：ch14\14.9.html）

```
<!DOCTYPE html>
<html>
<head>
<title>块元素</title>
<style>
.big{
width:800px;
height:105px;
background-image:url(07.jpg);
}
a{
font-size:12px;
display:block;
width:100px;
height:20px;
line-height:20px;
background-color:#F4FAFB;
text-align:center;
```

```
text-decoration:none;
border-bottom:1px dotted #6666FF;
color:black;
}
a:hover{
font-size:13px;
display:block;
width:100px;
height:20px;
line-height:20px;
text-align:center;
text-decoration:none;
color:green;
}
</style>
</head>
<body>
<div class="big">
<p>
<a href="#">管理应用</a><a href="#">财务管理</a><a href="#">在线管理</a>
<a href="#">客户关系管理</a><a href="#">一体化管理</a>
</p>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-9 所示，可以看到左边显示了一个导航栏，右边显示了一张图片。其导航栏就是以块元素形式显示。

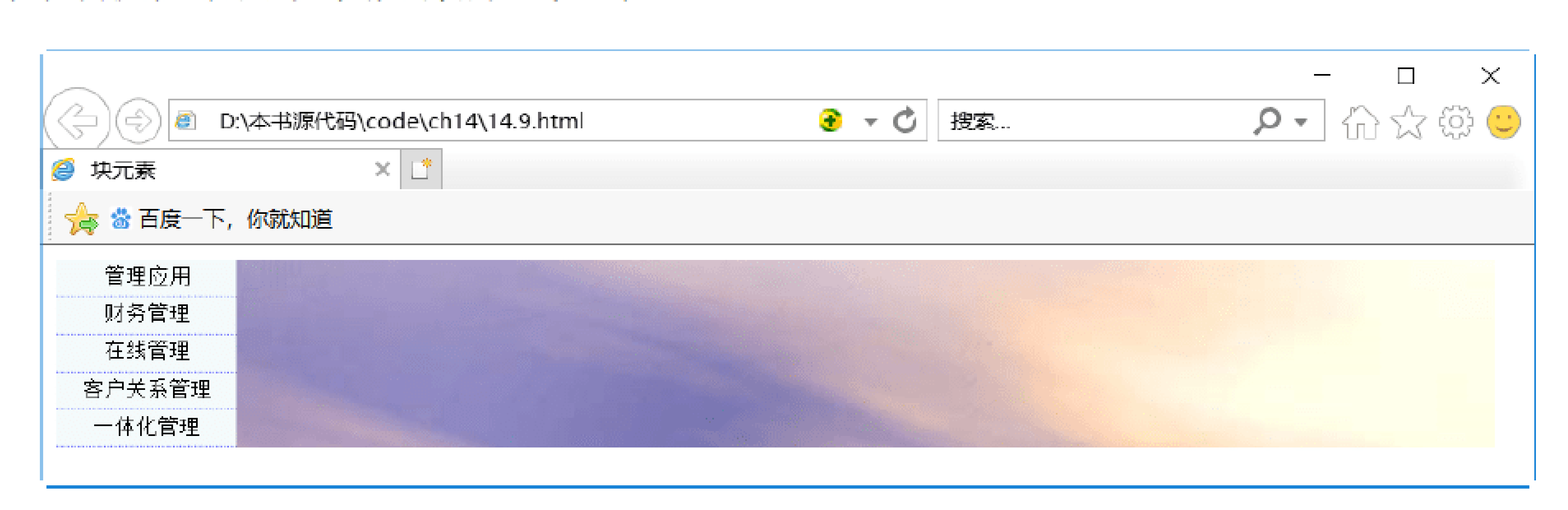


图 14-9　块元素显示

14.5.2　行内元素

当 display 的值被设置为 inline 时，可以把元素设置为行内元素，并在浏览器中同一行显示。inline 元素决定与其他 HTML 元素在同一行上，其行高、顶部和底部边距可以改变，而宽度是不可以改变的。

【例 14.10】（实例文件：ch14\14.10.html）

```
<!DOCTYPE html>
<html>
<head>
<title>行内元素</title>
<style type="text/css">
.hang {
display:inline;      //使段落生出行内框
}
</style>
</head>
<body>
<div>
<a href="#" class="hang">这是a标签</a>
<span class="hang">这是span标签</span>
<strong class="hang">这是strong标签</strong>
<img  class="hang" src=6.jpg/>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-10 所示，可以看到页面显示 44 个 HTML 元素，都在同一行显示，包括超级链接、文本信息和图片。

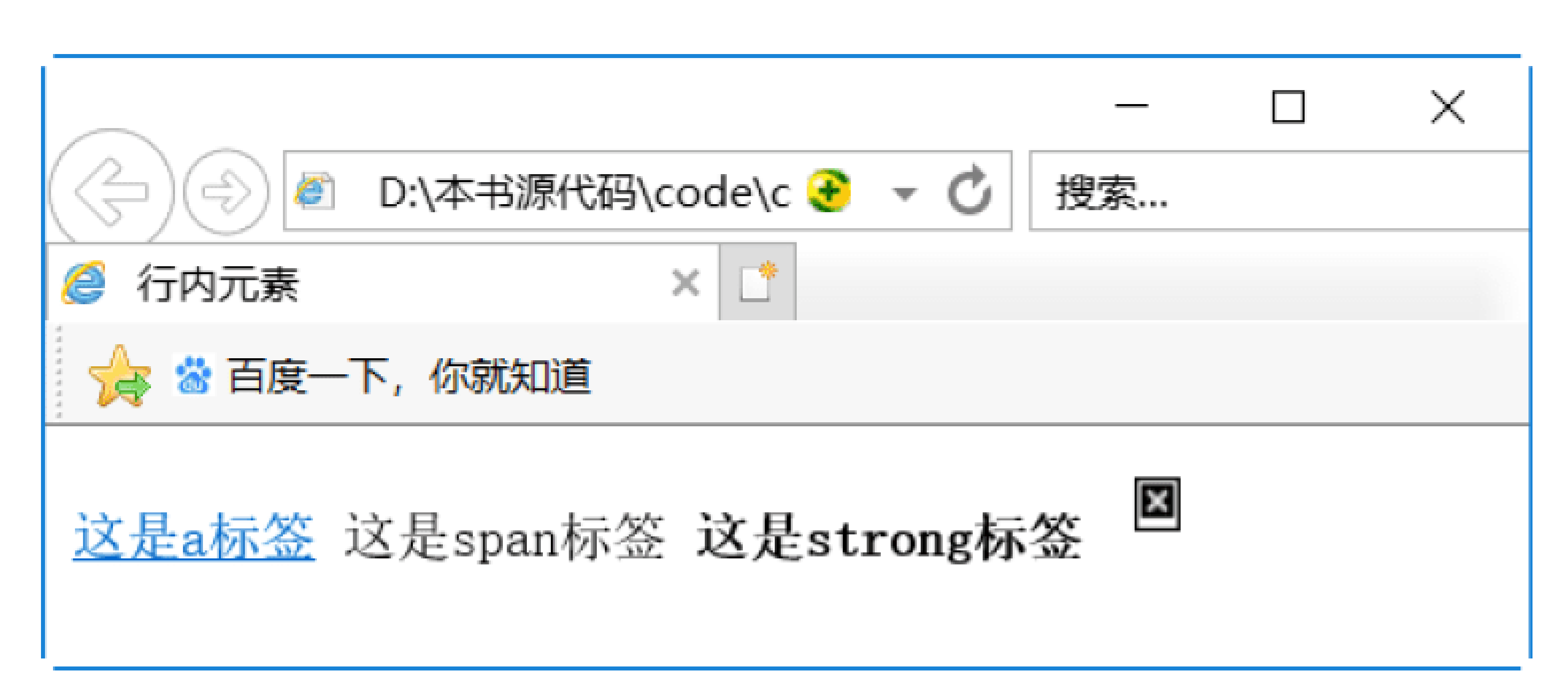


图 14-10　行内显示

14.6　综合实例——定位布局新闻

一个美观大方的页面，必然是一个布局合理的页面。左右布局是网页中比较常见的一种方式，即根据信息种类不同将信息分别在当前页面左右侧显示。本实例将利用前面学习的知识创建一个左右布局的新闻页面。

具体步骤如下：

01 分析需求。

首先需要将整个页面分为左右两个模块：左模块放置一类信息；右模块放置一类信息，可以设置其宽度和高度。

02 创建 HTML 页面，实现基本列表。

创建 HTML 页面，同时用 DIV 在页面中划分左边 div 层和右边 div 层两个区域，并且将信息放入到相应的 div 层中，注意 div 层内引用 CSS 样式名称。

```
<!DOCTYPE html>
<html>
<head>
<title>布局</title>
</head>
<body>
<center>
<div class="big">
  <p class=pp>女人</p>
<div class="left">
  <h1>女人</h1>
  <p> •男人幸福告白：女人的性感与年龄成正比09:59 </p>
  <p> •六类食物能有效对抗紫外线11:15 </p>
  <p> •打造夏美人 受OL追捧的清爽发型10:05 </p>
  <p> •美丽帮帮忙：别让大油脸吓跑男人09:47 </p>
  <p> •简约雪纺清凉衫 百元搭出欧美范儿14:51 </p>
  <p> •花边连衣裙超勾人 7月穿搭出新意11:04 </p>
</div>
<div class="right">
  <h1>健康</h1>
  <p> •女性养生：让女人老得快的10个原因19:18 </p>
  <p> •养生盘点：喝豆浆的九大好处和七大禁忌09:14</p>
  <p> •养生警惕：14个护肤心理“错”觉19:57</p>
  <p> •柿子番茄骨汤 8种营养师最爱的食物15:16</p>
  <p> •夏季养生指南：“夫妻菜”宜常吃10:48 </p>
  <p> •10条食疗养生方法，居家宅人的养生经13:54 </p>
</div>
</div>
</center>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 14-11 所示，可以看到页面显示了两个模块，分别是“女人”和“健康”，二者上下排列。

03 添加 CSS 代码，修饰整体样式和 div 层。

```
<style>
```

```
* {
    padding:0px;
    margin:0px;
}
body {
    font:"宋体";
    font-size:12px;
}
.big{
    width:570px;
height:210px;
border:#C1C4CD 1px solid;
}
</style>
```

在 IE 11.0 中浏览效果如图 14-12 所示，可以看到页面比较原来字体变小，并且大的 DIV 显示了边框。

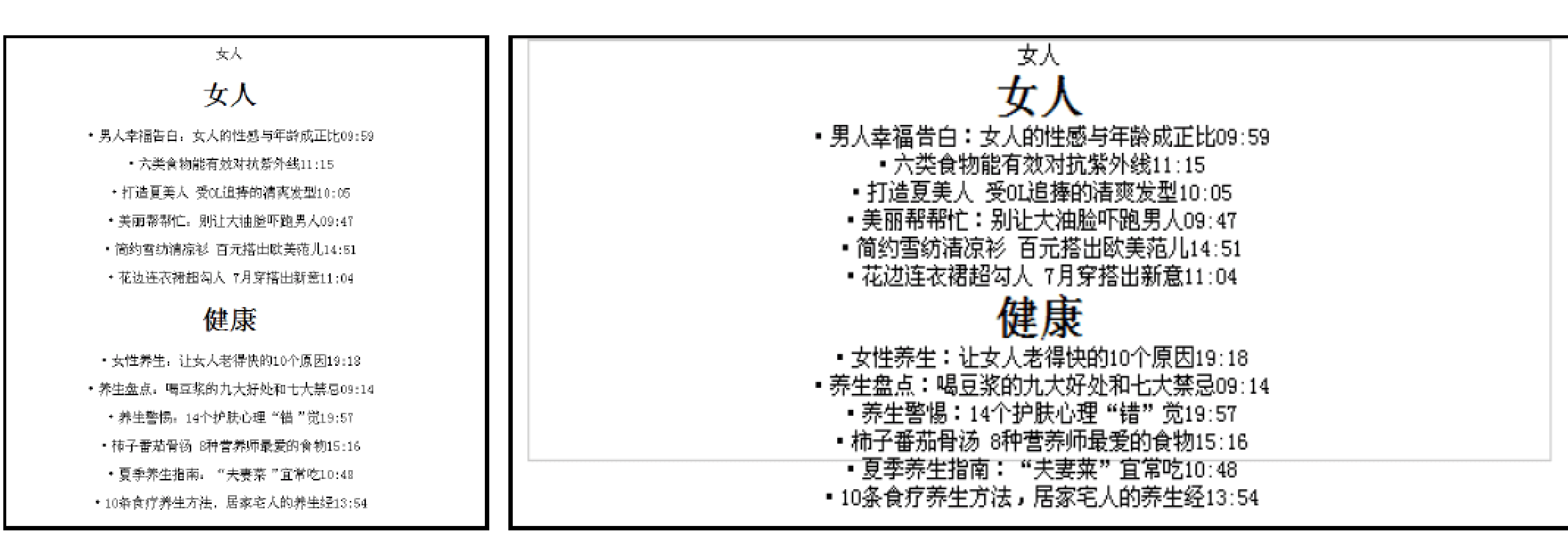


图 14-11　上下排列　　　　图 14-12　修饰整体样式

04 添加 CSS 代码，设置两个层左右并列显示。

```
.left{
    width:280px;
    float:right; //设置右边悬浮
    border:#C1C4CD 1px solid;
    }.right{
    width:280px;
    float:left;//设置左边悬浮
    margin-left:6px;
    border:#C1C4CD 1px solid;
}
```

在 IE 11.0 中浏览效果如图 14-13 所示，可以看到页面中文本信息左右并列显示，但字体没有发生变化。

05 添加 CSS 代码，定义文本样式。

```
h1{
    font-size:14px;
    padding-left:10px;
    background-color:#CCCCCC;
    height:20px;
    line-height:20px;
    }
p{
margin:5px;
line-height:18px;
color:#2F17CD;
}
.pp{
width:570px;
text-align:left;
height:20px;
background-color:D5E7FD;
position:relative;
left:-3px;
top:-3px;
font-size:16px;
text-decoration:underline;
}
```

在 IE 11.0 中浏览效果如图 14-14 所示，可以看到页面中文本信息左右并列显示，其字体颜色为蓝色，行高为 18 像素。

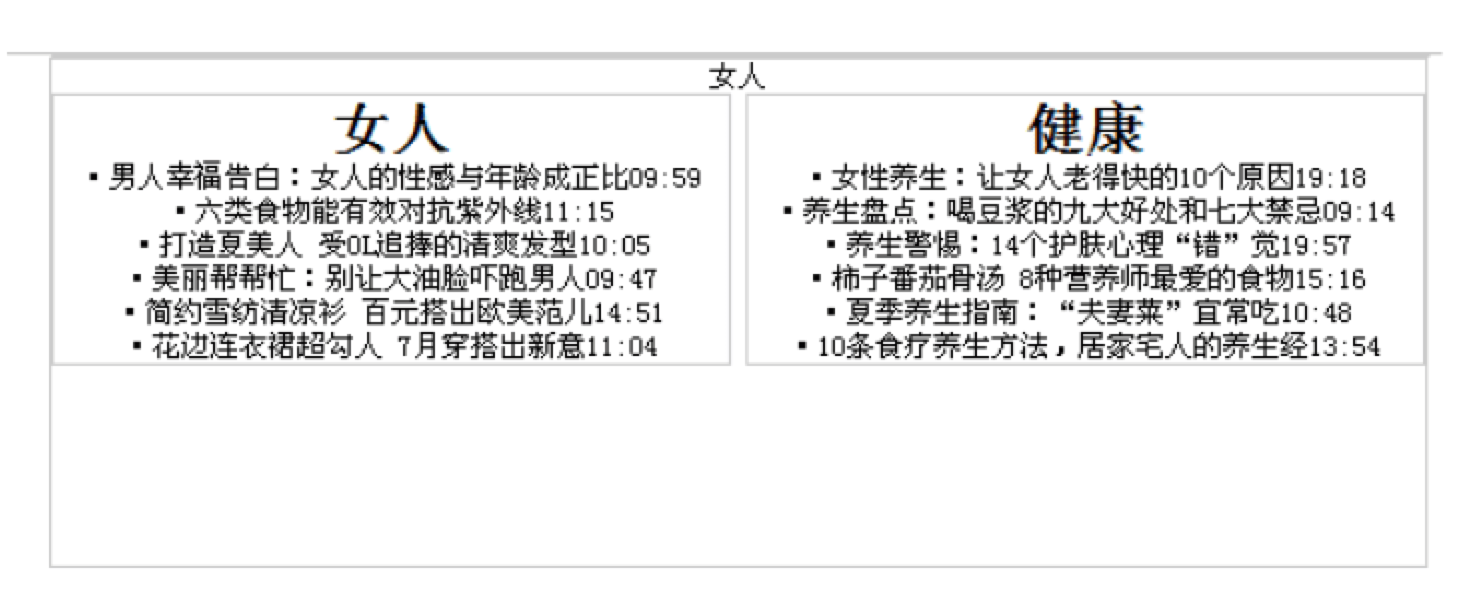


图 14-13 设置左右悬浮



图 14-14 文本修饰样式

14.7 专家解惑

1. 块元素和行内元素的概念是什么?有哪些特点？

块级元素和行内元素是布局最基本的两种元素，常见块级元素有 div、p、form、ul、ol、li 等，常见行内元素有 span、strong、em 等。

块级元素会独占一行，对应于 display:block，可以设置 width、height、margin、padding 属性；行内元素不独占一行，对应于 display:inline，相邻行内元素会排列在同一行里，直到排不下才换行，设置 width、height 属性无效，而 margin 和 padding 属性只对设置水平方向的 right 和 left 有效。可以通过修改 display 属性来切换块级元素和行内元素。

display:inline-block 是行内的块级元素，拥有块级元素的特点，可以设置 width、height、margin、padding 值，但又可以与其他行内元素排在同一行里。

2. 当设置多个 div 并列时，为什么会撑破整个布局？

很多时候，尤其是容器内有平行布局，如两三个 float 的 div 时，宽度很容易出现问题。在 IE 中，外层的宽度会被内层更宽的 div 挤破。一定要用 Photoshop 或 Firework 量取像素级的精度。

第15章　CSS3盒子和DIV布局

在传统网页设计中，为了保证页面元素位置，经常使用表格来完成。表格起到了定位和布局的作用。由于表格布局的局限性，目前往往采用 DIV+CSS3 方式。CSS3 中提出了盒子模型和新增盒子模型来完成对元素的直接定位，即能够为页面元素定义边框，并修饰内容距离，从而优化文本内容的显示效果。本章将学习 CSS3 盒子和 DIV 布局的方法和技巧。

15.1　认识 div 层

使用 DIV 进行网页排版是现在流行的一种趋势，如使用 CSS3 属性，可以轻易设置 DIV 位置，演变出多种不同的布局方式。

15.1.1　层在 HTML 布局应用

<div>标记是一个区块容器标记，在<div></div>标记中可以放置其他的一些 HTML 元素，如段落<p>、标题<h1>、表格<table>、图片<img>和表单等。然后使用 CSS3 相关属性对 div 容器标记中的元素作为一个独立对象进行修饰，这样就不会影响其他 HTML 元素。

【例 15.1】（实例文件：ch15\15.1.html）

```
<!DOCTYPE html>
<html>
<head>
<title>div 层</title>
<style type="text/css">
div{
   font-size:18px;
   font-weight:bolder;
  font-family:“幼圆”;
   color:#FF0000;
   background-color:#eeddcc;
   text-align:center;
   width:300px;
   height:100px;
   border:1px #992211 dotted;
```

```
    }
    </style>
    </head>
    <body>
    <center>
    <div>
      这是div层
    </div>
    </center>
    </body>
    </html>
```

上面例子通过 CSS 对 div 块控制，绘制了一个 div 容器，容器中放置了一段文字。在 IE 11.0 中浏览效果如图 15-1 示，可以看到一个矩形方块的 div 层，居中显示，字体显示为红色，边框为浅红色，背景色为浅黄色。

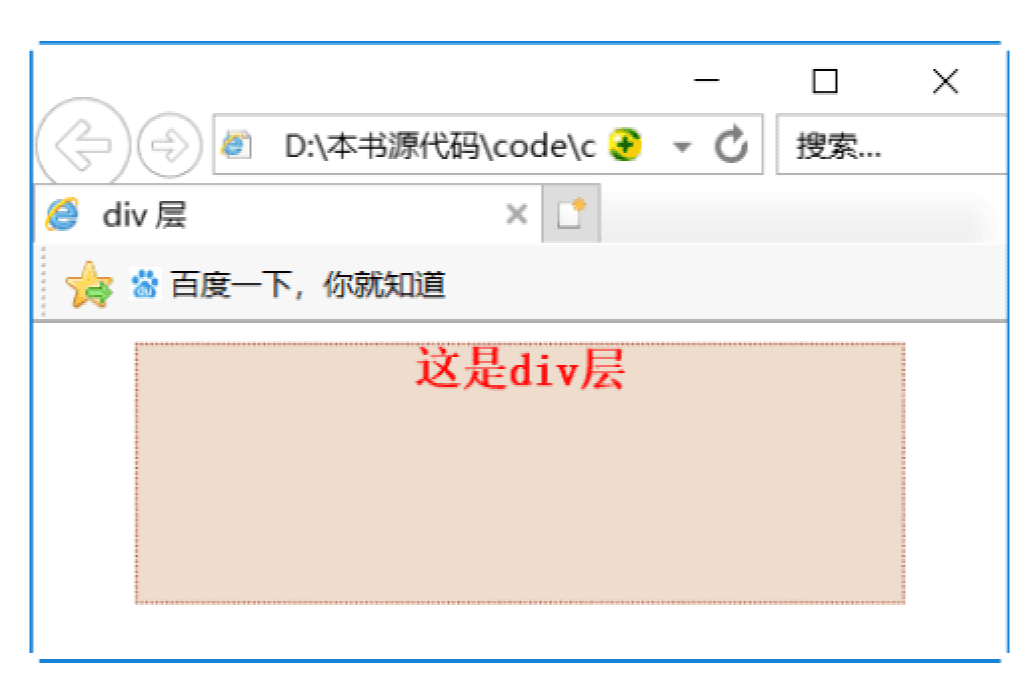


图 15-1　div 层显示

15.1.2　div 和 span 区别

对于初学者而言，div 和 span 两个标记常常混淆，因为大部分的 div 层都可以使用 span 标记代替，并且其运行效果完全一样，例如上一节中的例子，就可以使用 span 标记代替 div 标记，其运行效果完全保持一致。可以这样说，在使用区块对 HTML 元素进行包含方面，div 和 span 标记的作用基本一样。

div 和 span 标记二者的区别在于，div 是一个块级元素，其包含的元素会自动换行。span 标记是一个行内标记，其前后都不会发生换行。div 标记可以包含 span 标记元素，但 span 标记一般不包含 div 标记。

【例 15.2】（实例文件：ch15\15.2.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <title>div与span的区别</title>
```

```
  </head>
  <body>
    <p>div自动分行：</p>
    <div><b>宁静</b></div>
    <div><b>致远</b></div>
    <div><b>明治</b></div>
    <p>span同一行：</p>
    <span><b>老虎</b></span>
    <span><b>狮子</b></span>
    <span><b>老鼠</b></span>
  </body>
  </html>
```

在 IE 11.0 中浏览效果如图 15-2 示，可以看到 div 层所包含的元素，进行自动换行，而对于 span 标记，3 个 HTML 元素在同一行显示。

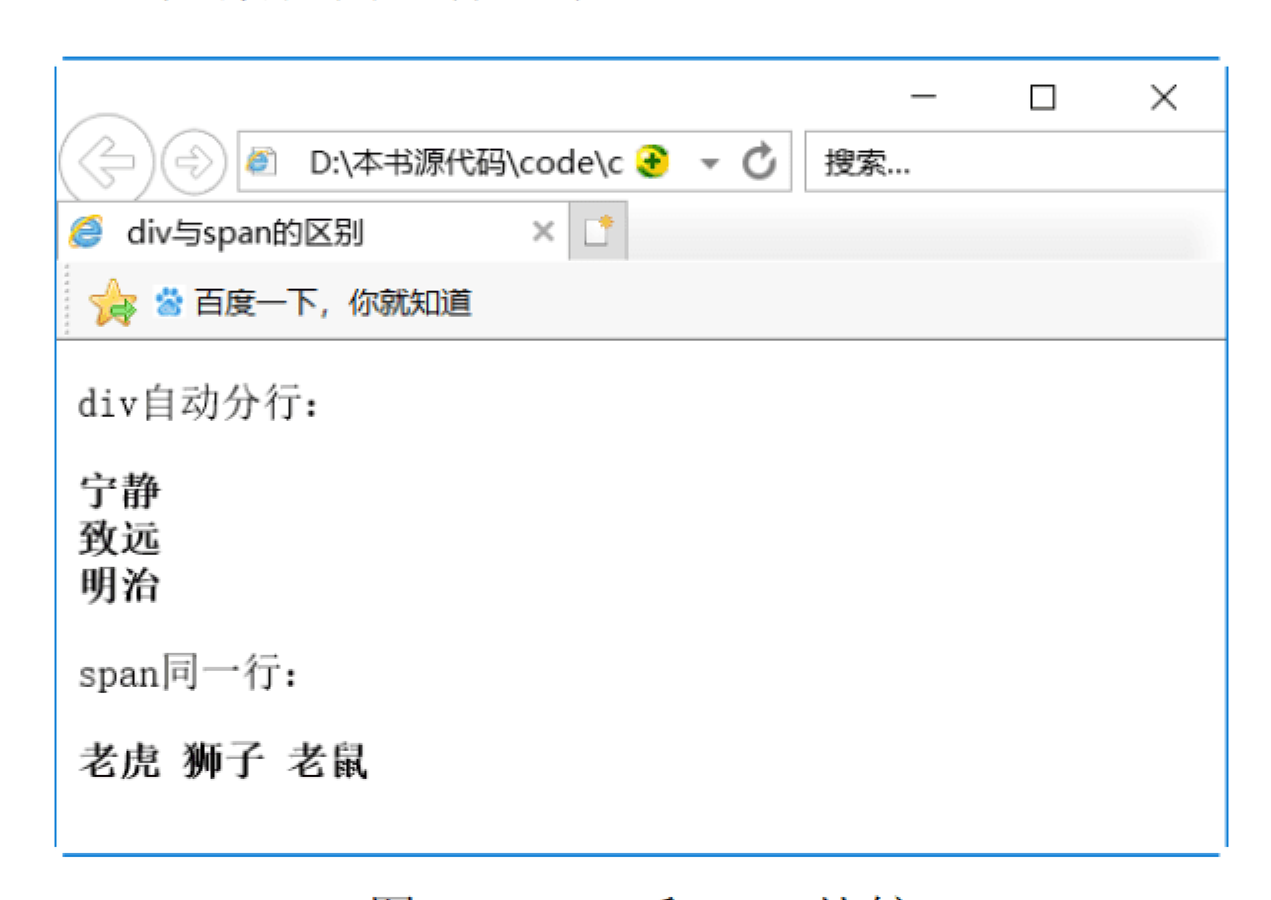


图 15-2　div 和 span 比较



在网页设计中，对于较大的块可以使用 div 完成，而对于具有独特样式的单独 HTML 元素，可以使用 span 标记完成。

15.2　盒子模型

将网页上每个 HTML 元素都认为是长方形的盒子，是网页设计上的一大创新。在控制页面方面，盒子模型起着至关重要的作用，熟练掌握盒子模型及其盒子模型各个属性，是控制页面中每个 HTML 元素的前提。

15.2.1　什么是盒子模型

CSS3 中，所有的页面元素都包含在一个矩形框内，称为盒子。盒子描述了元素及其属性在页面布局中所占的空间大小，因此盒子可以影响其他元素的位置及大小。例如，页面中第一

个盒子为 10 像素，那么下一个盒子就处于离顶部 10 像素距离的位置。如果第一个盒子增加为 20 像素，则下一个盒子就要再下移 10 像素，而整个页面就是由这些个大大小小，但不会重叠的盒子形成的。

盒子模型是由 margin（边界）、border（边框）、padding（空白）和 content（内容）几个属性组成。此外，在盒子模型中还具备高度和宽度两个辅助属性。盒子模型如图 15-3 所示。

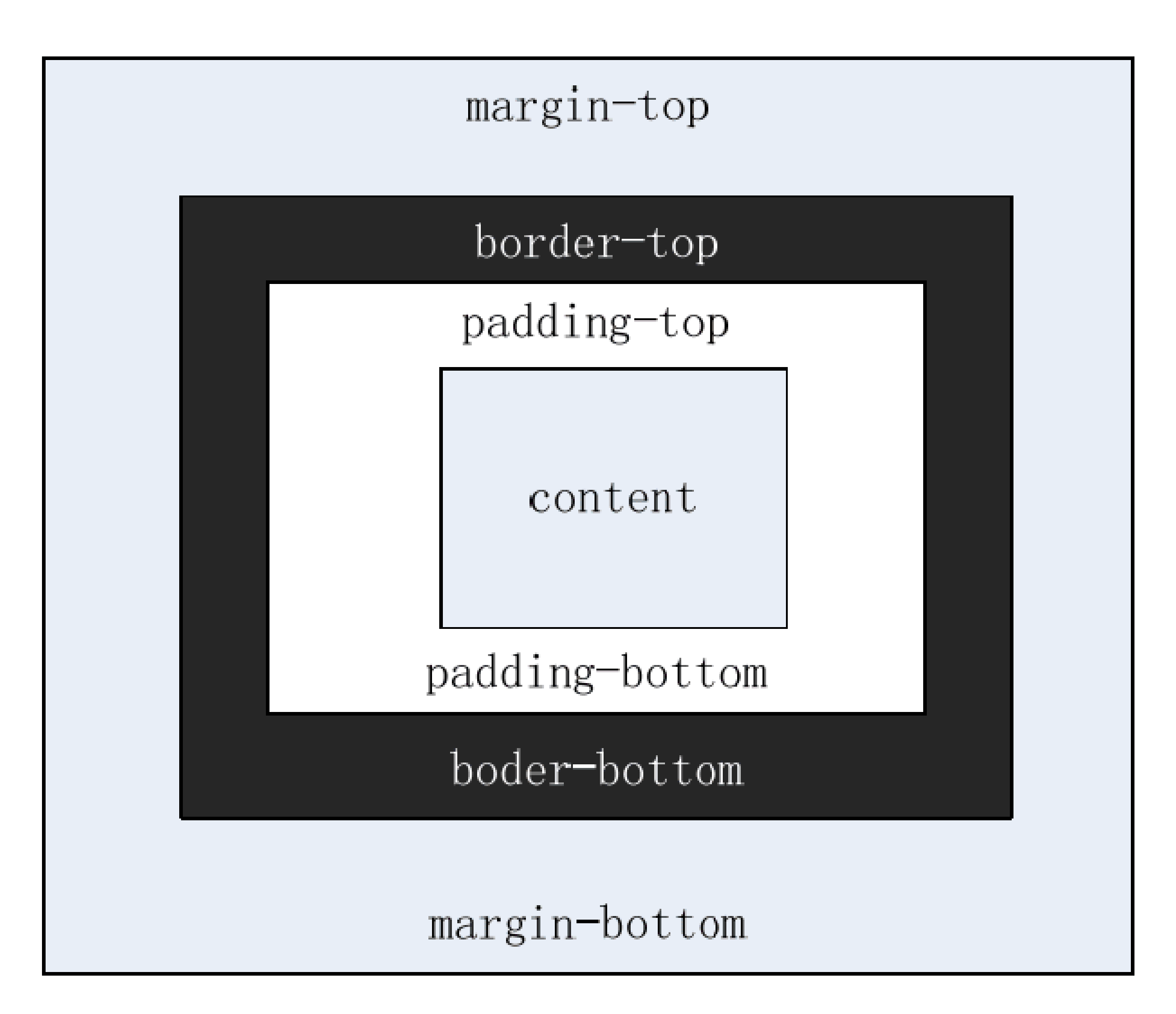


图 15-3　盒子模型效果图

从图 15-3 中可以看出，盒子模型包含如下 4 部分。

（1）content（内容）：是盒子模型中必需的一部分，内容可以是文字、图片等元素。

（2）padding（空白）：也称内边距或补白，用来设置内容和边框直接的距离。

（3）border（边框）：可以设置内容边框线的粗细、颜色和样式等，前面已经介绍过。

（4）margin（边界）：外边距，用来设置内容与内容之间的距离。

对于这些属性可以把它转移到日常生活中的盒子（箱子）上来理解，日常生活中所见的盒子也就是能装东西的一种箱子，也具有这些属性，所以称其为盒子模式。内容（content）就是盒子里装的东西；填充（padding）就是怕盒子里装的东西（贵重的）损坏而添加的泡沫或其他抗震的辅料；边框（boder）就是盒子本身；至于边界（margin）则说明盒子摆放的时候不能全部堆在一起，要留出一定的空隙保持通风，同时也为了方便取出。



在网页设计上，内容常指文字、图片等元素，但是也可以是小盒子（DIV 嵌套），与现实生活中盒子不同的是，现实生活中的东西一般不能大于盒子，否则盒子会被撑坏的，而 CSS 盒子具有弹性，里面的东西大过盒子本身最多把它撑大，但不会损坏。

一个盒子的实际高度（宽度）是由 content+padding+border+margin 组成的。在 CSS3 中，可以通过设置 width 和 height 来控制 content 的大小，并且对于任何一个盒子，都可以分别设置 4 条边的 border、padding 和 margin。

15.2.2 border 边框

border 边框是内边距和外边距的分界线，可以分离不同的 HTML 元素，boder 的外围是元素的最外围。在网页设计中，如果计算元素的宽和高，则需要把 border 计算在内。

【例 15.3】（实例文件：ch15\15.3.html）

```
<!DOCTYPE html>
<html>
<head>
<title>border边框</title>
<style type="text/css">
.div1{
    borde-widthr:10px;
    border-color:#ddccee;
    border-style:solid;
    width:410px;
}
.div2{
    border-width:1px;
    border-color:#adccdd;
    border-style:dotted;
    width:410px;
}
.div3{
    border-width:1px;
    border-color:#457873;
    border-style:dashed;
    width:410px;
}
</style>
</head>
<body>
<div class="div1">
    这是一个宽度为10px的实线边框。
</div><br /><br />
<div class="div2">
    这是一个宽度为1px的虚线边框。
</div> <br /><br />
<div class="div3">
    这是一个宽度为1px的点状边框。
</div>
</body>
```

```
</html>
```

在 IE 11.0 中浏览效果如图 15-4 示，可以看到显示了 3 个不同风格的盒子，第一个盒子边框宽度为 10 像素，边框样式为实线，颜色为紫色；第二个盒子边框宽度为 1 像素，边框样式是虚线边框，颜色为浅绿色；第三个盒子边框宽度为 1 像素，边框样式是点状边框，颜色为绿色。

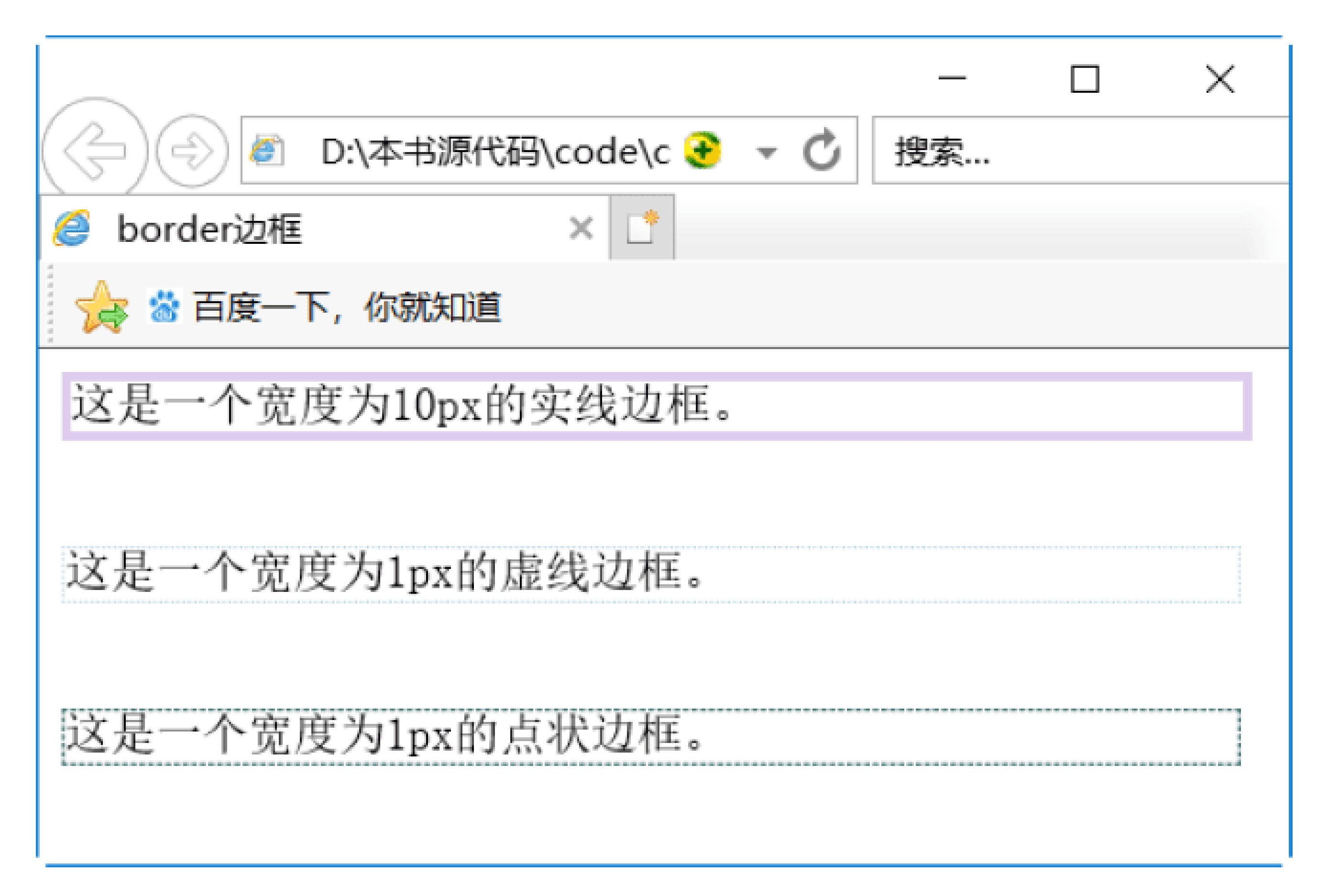


图 15-4　设置盒子边框



这里需要注意的是，在给元素设置 background-color 背景色时，IE 作用的区域为 content+padding，而 Firefox 则是 content+padding+border，这点在 border 为粗虚线时特别明显。

15.2.3　padding 内边距

在 CSS3 中，可以设置 padding 属性定义内容与边框之间的距离，即内边距的距离。

语法格式如下：

```
padding : length
```

padding 属性值可以是一个具体的长度，也可以是一个相对于上级元素的百分比，但不可以使用负值。当设置值为百分数时，百分数值是相对于其父元素的 width 计算的，这一点与外边距一样。所以，如果父元素的 width 改变，则其值也会改变。

padding 属性能为盒子定义上、下、左、右间隙的宽度，也可以单独定义各方位的宽度。常用形式如下：

```
padding :padding-top | padding-right | padding-bottom | padding-left
```

如果提供 4 个参数值，则将按顺时针的顺序作用于四边。如果只提供 1 个参数值，则将用于全部的四条边；如果提供 2 个参数值，则第一个作用于上下两边，第 2 个作用于左右两边；如果提供 3 个参数值，则第 1 个作用于上边，第 2 个作用于左、右两边，第 3 个作用于下边。

其具体含义如表 15-1 所示。

表 15-1　padding 属性子属性

属性	描述
padding-top	设置上间隙
padding-bottom	设置下间隙
padding-left	设置左间隙
padding-right	设置右间隙

【例 15.4】（实例文件：ch15\15.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>padding</title>
<style type="text/css">
.wai{
    width:400px;
    height:250px;
    border:1px #993399 solid;
}
img{
    max-height:120px;
    padding-left:50px;
    padding-top:20px;
}
</style>
</head>
<body>
<div class="wai">
  <img src="13.jpg" />
  <p>这张图片的左内边距是50px，顶内边距是20px</p>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 15-5 示，可以看到一个 div 层中显示了一张图片，该图片可以看作是一个盒子模型，并定义了图片的左内边距和上内边距的效果。内边距其实是对象 img 和外层 DIV 之间的距离。

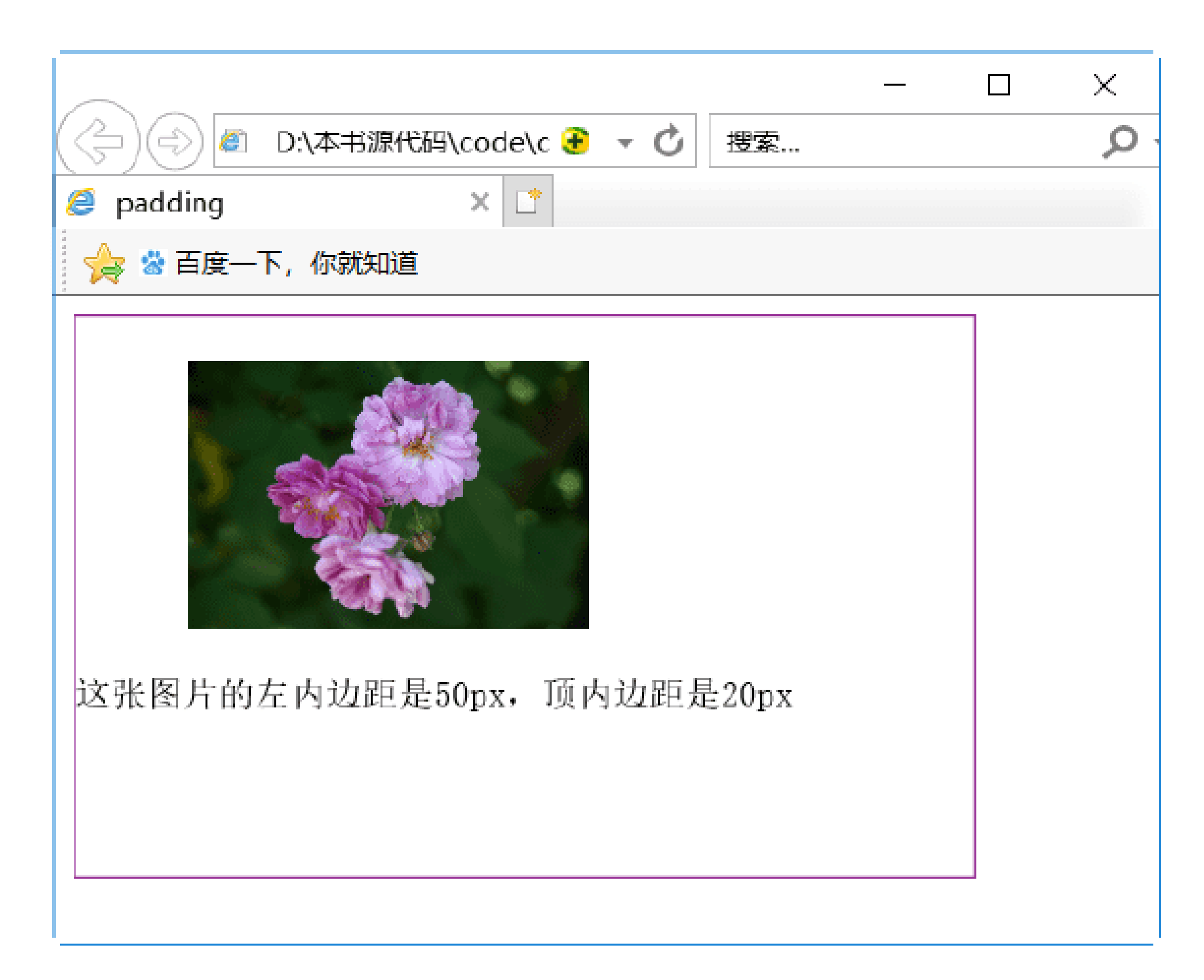


图 15-5　设置内边距

15.2.4　margin 外边距

margin 边界用来设置页面中元素和元素之间的距离，即定义元素周围的空间范围，是页面排版中一个比较重要的概念。

语法格式如下：

```
margin : auto | length
```

其中 auto 表示根据内容自动调整；length 表示由浮点数字和单位标识符组成的长度值或百分数，百分数是基于父对象的高度。对于内联对象来说，左右外延边距可以是负数值。

margin 属性包含的 4 个子属性控制一个页面元素四周的边距样式，如表 15-2 所示。

表 15-2　margin 属性子属性

属性	描述
margin-top	设置上边距
margin-bottom	设置下边距
margin-left	设置左边距
margin-right	设置右边距

各子属性的属性值同样可以是一个确定的长度，也可以是一个百分比，该百分比相对于其上级元素的宽度（width）。

在给 margin 设置值时，如果提供 4 个参数值，则将按顺时针的顺序作用于四边。如果只提供 1 个参数值，则将作用于全部的四条边；如果提供 2 个参数值，则第 1 个作用于上下两边，第 2 个作用于左右两边；如果提供 3 个参数值，则第 1 个作用于上边，第 2 个作用于左右两边，

第 3 个作用于下边。

如果希望很精确地控制块的位置，则需要对 margin 有更加深入地了解。margin 设置可以分为行内元素块之间设置、非行内元素块之间设置和父子块之间设置。

1. 行内元素 margin 设置

【例 15.5】（实例文件：ch15\15.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>行内元素设置margin</title>
<style type="text/css">
span{
    background-color:#a2d2ff;
    text-align:center;
    font-family:"幼圆";
    font-size:12px;
    padding:10px;
    border:1px #ddeecc solid;
}
span.left{
    margin-right:20px;
    background-color:#a9d6ff;
}
span.right{
    margin-left:20px;
    background-color:#eeb0b0;
}
</style>
</head>
<body>
  <span class="left">行内元素1</span><span class="right">行内元素2</span>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 15-6 所示，可以看到一个蓝色盒子和红色盒子，二者之间的距离使用 margin 设置，其距离是左边盒子的右边距 margin-right 加上右边盒子的左边距 margin-left。

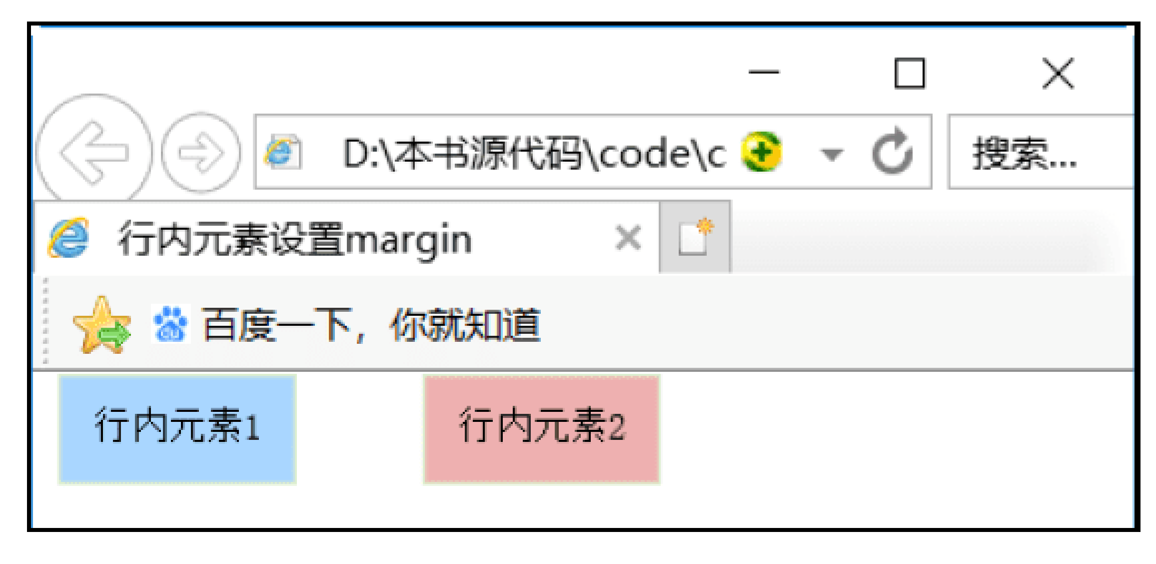


图 15-6　设置外边距

2. 非行内元素块之间 margin 设置

如果不是行内元素，而是产生换行效果的块级元素，情况就会发生变化。两个换行块级元素之间的距离不再是 margin-bottom 和 margin-top 的和，而是两者中的较大者。

【例 15.6】（实例文件：ch15\15.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>块级元素的margin</title>
<style type="text/css">
h1{
    background-color:#ddeecc;
    text-align:center;
    font-family:"幼圆";
    font-size:12px;
    padding:10px;
    border:1px #445566 solid;
    display:block;
}
</style>
</head>
<body>
<h1 style="margin-bottom:50px;">距离下面块的距离</h1>
<h1 style="margin-top:30px;">距离上面块的距离</h1>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 15-7 所示，可以看到两个 h1 盒子，二者上下之间存在距离，其距离为 margin-bottom 和 margin-top 中较大的值，即 50 像素。如果修改下面 h1 盒子元素的 margin-top 为 40 像素，则会发现执行结果没有任何变化；如果修改其值为 60 像素，则会发现下面的盒子向下移动 10 个像素。

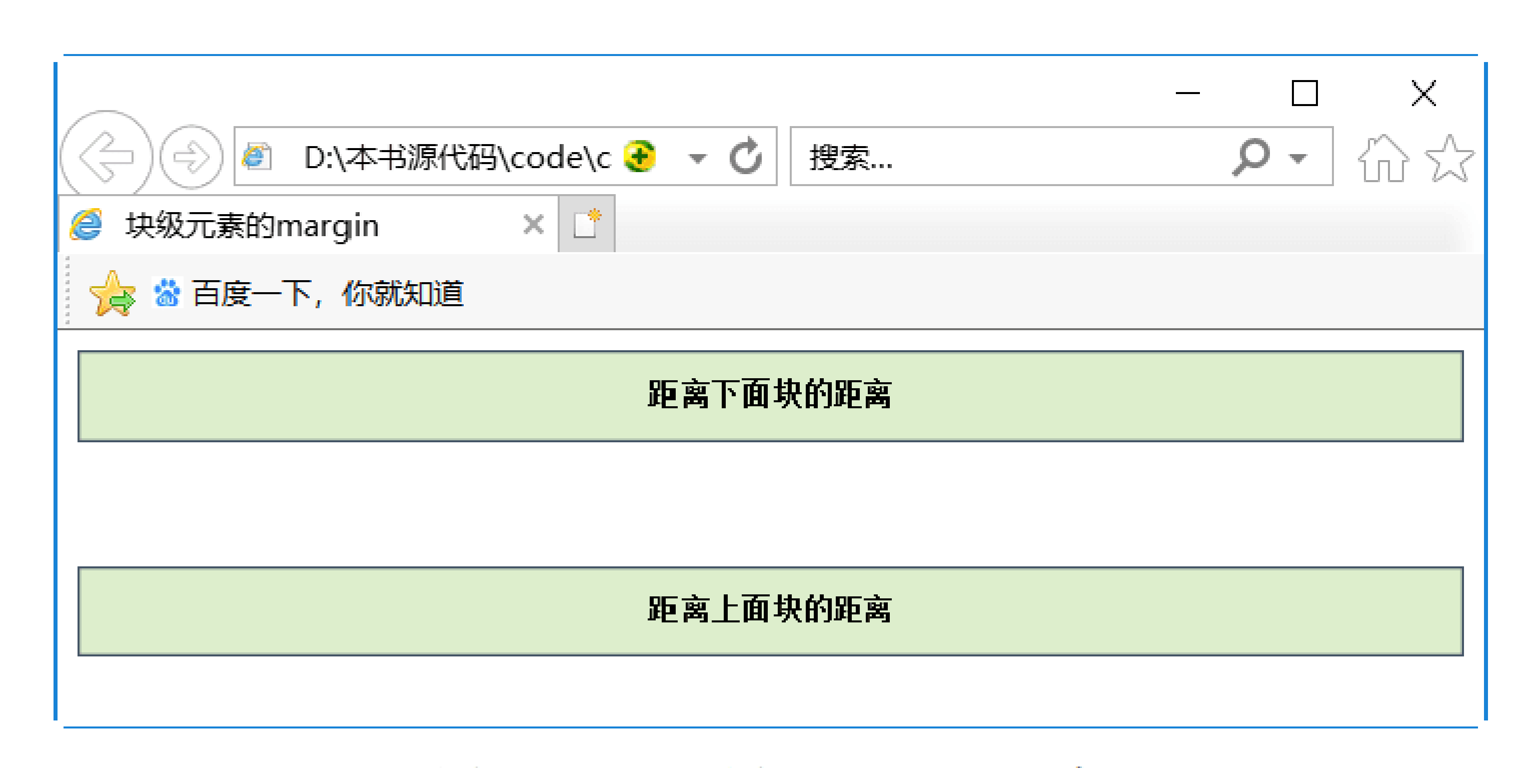


图 15-7　设置上下 margin 距离

3. 父子块之间 margin 设置

当一个 div 块包含在另一个 div 块中间时，二者便会形成一个典型的父子关系。其中子块的 margin 设置将会以父块的 content 为参考。

【例 15.7】（实例文件：ch15\15.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title>包含块的margin</title>
<style type="text/css">
div{
   background-color:#fffebb;
   padding:10px;
   border:1px solid #000000;
}
h1{
   background-color:#a2d2ff;
   margin-top:0px;
   margin-bottom:30px;
   padding:15px;
   border:1px dashed #004993;
   text-align:center;
   font-family:"幼圆";
   font-size:12px;
}
</style>
</head>
<body>
<div >
  <h1>子块div</h1>
</div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 15-8 所示，可以看到子块 h1 盒子距离父 div 下边界为 40 像素（子块 30 像素的外边距加上父块 10 像素的内边距），其他 3 边距离都是父块的 padding 距离，即 10 像素。

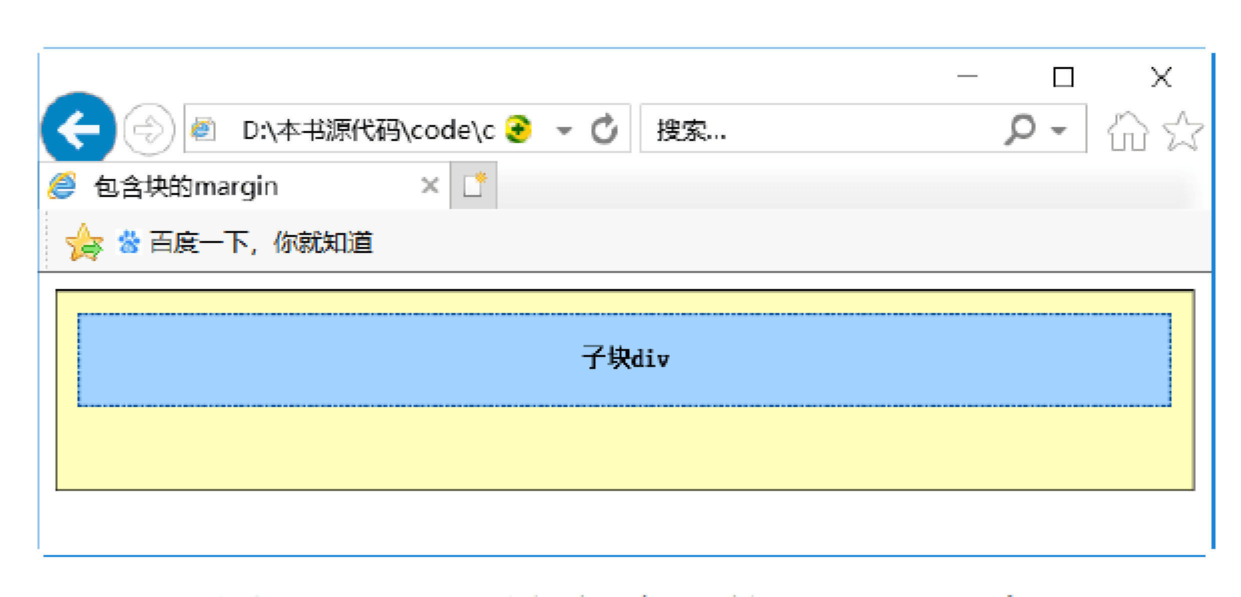


图 15-8　设置包括盒子的 margin 距离

在上例中，如果设置了父元素的高度 height 值，并且父块高度值小于子块的高度加上 margin 的值，则 IE 浏览器会自动扩大，保持子元素的 margin-bottom 的空间及父元素的 padding-bottom。而 Firefox 就不会这样，会保证父元素 height 高度的完全吻合，而子元素将超过父元素的范围。

当将 margin 设置为负数时，会使得被设为负数的块向相反的方向移动，甚至覆盖在另外的块上。

15.3　CSS3 新增弹性盒模型

CSS3 引入了新的盒模型处理机制，即弹性盒模型。该模型决定元素在盒子中的分布方式以及如何处理盒子的可用空间。通过弹性盒模型，可以轻松地设计出自适应浏览器窗口的流动布局或自适应字体大小的弹性布局。

CSS3 为了弹性盒模型，新增了 8 个属性，如表 15-3 所示。

表 15-3　CSS3 新增盒子模型属性

属性	说明
box-orient	定义盒子分布的坐标轴
box-align	定义子元素在盒子内垂直方向上的空间分配方式
box-direction	定义盒子的显示顺序
box-flex	定义子元素在盒子内的自适应尺寸
box-flex-group	定义自适应子元素群组
box-lines	定义子元素分布显示
box-ordinal-group	定义子元素在盒子内的显示位置
box-pack	定义子元素在盒子内的水平方向上的空间分配方式

15.3.1　盒子布局取向 box-orient

box-orient 属性用于定义盒子元素内部的流动布局方向。

语法格式如下：

```
box-orient:horizontal | vertical | inline-axis | block-axis | inherit
```

其参数值含义如表 15-4 所示。

表 15-4 box-orient 属性值

属性值	说明
horizontal	盒子元素从左到右在一条水平线上显示它的子元素
vertical	盒子元素从上到下在一条垂直线上显示它的子元素
inline-axis	盒子元素沿着内联轴显示它的子元素
block-axis	盒子元素沿着块轴显示它的子元素

弹性盒模型是 W3C 标准化组织于 2009 年发布的，目前还没有主流浏览器对其支持，不过采用 Webkit 和 Mozilla 渲染引擎的浏览器都自定义了一套私有属性，用来支持弹性盒模型。下面代码中会存在一些 Firefox 浏览器的私有属性定义。

【例 15.8】（实例文件：ch15\15.8.html）

```
<!DOCTYPE html>
<html>
<head>
<title>
box-orient
</title>
<style>
div{height:50px;text-align:center;}
.d1{background-color:#F6F;width:180px;height:300px}
.d2{background-color:#3F9;width:300px;height:300px}
.d3{background-color:#FCd;width:180px;height:300px}
body{
    display:box;/*标准声明，盒子显示*/
    display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
    orient:horizontal;/*定义元素为盒子显示*/
    -mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
    box-orient:horizontal;/*CSS3标准化设置*/
}
</style>
</head>
<body>
<div class=d1>左侧布局</div>
<div class=d2>中间布局</div>
<div class=d3>右侧布局</div>
</body>
</html>
```

上面代码中，CSS 样式首先定义了每个 div 层的背景色和大小，在 body 标记选择器中定义了 body 容器中元素以盒子模型显示，并使用 box-orient 定义元素水平并列显示。

在 Firefox 62.0 中浏览效果如图 15-9 所示，可以看到显示了 3 个层，3 个 div 层并列显示，

分别为“左侧布局”“中间布局”和“右侧布局”。

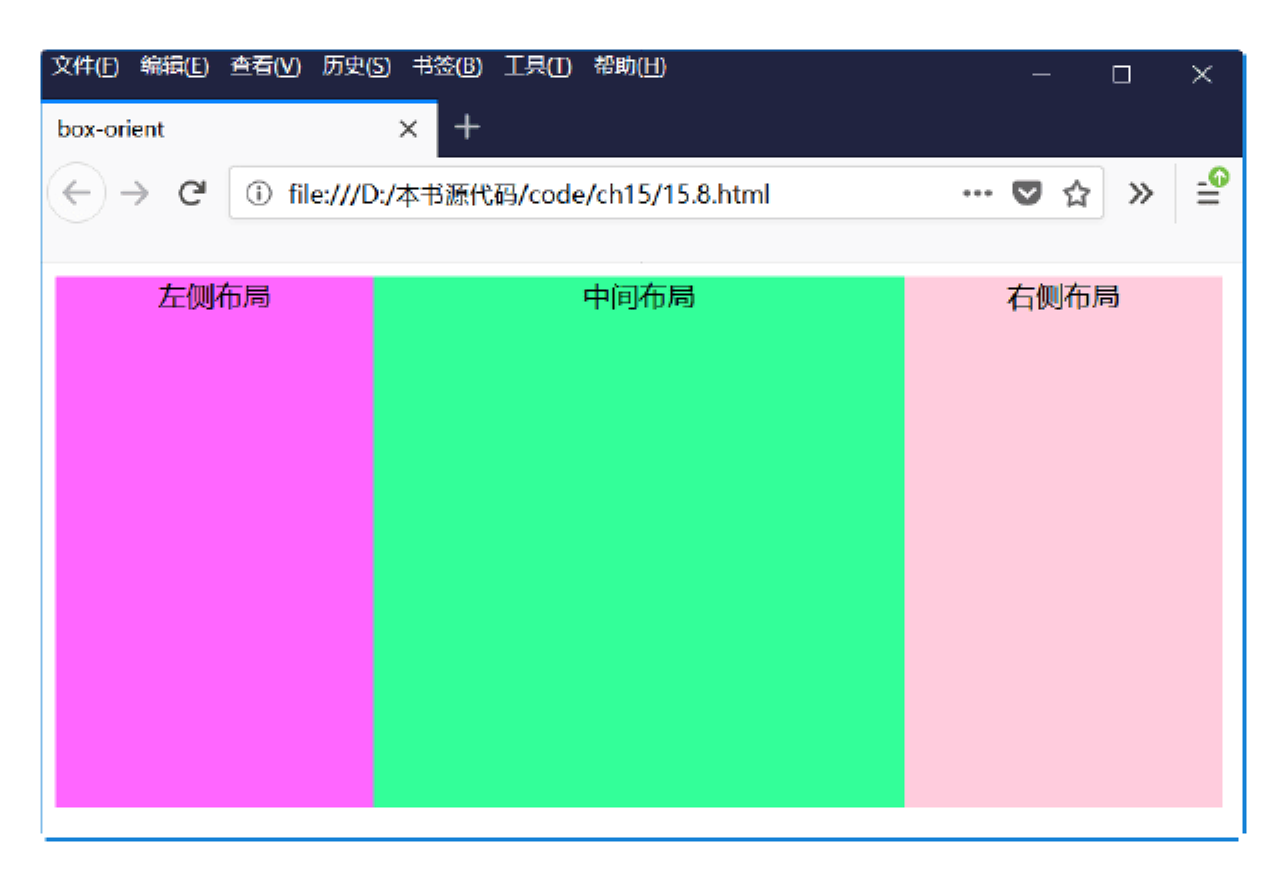


图 15-9　盒子元素水平并列显示

15.3.2　盒子布局顺序 box-direction

box-direction 是用来确定子元素的排列顺序，也可以说是内部元素的流动顺序。

语法格式如下：

```
box-direction:normal | reverse | inherit
```

其参数值如表 15-5 所示。

表 15-5　box-direction 属性值

属性值	说明
normal	正常显示顺序，如果盒子元素的 box-orient 属性值为 horizontal，则其包含的子元素按照从左到右的顺序显示，即每个子元素的左边总是靠近前一个子元素的右边；如果盒子元素的 box-orient 属性值为 vertical，则其包含的子元素按照从上到下的顺序显示
reverse	反向显示，盒子所包含的子元素的显示顺序将与 normal 相反
inherit	继承上级元素的显示顺序

【例 15.9】（实例文件：ch15\15.9.html）

```
<!DOCTYPE html>
<html>
<head>
<title>
box-direction
</title>
<style>
div{height:50px;text-align:center;}
```

```
.d1{background-color:#F6F;width:180px;height:300px}
.d2{background-color:#3F9;width:300px;height:300px}
.d3{background-color:#FCd;width:180px;height:300px}
body{
    display:box;/*标准声明，盒子显示*/
    display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
    orient:horizontal;/*定义元素为盒子显示*/
    -mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
    box-orient:horizontal;/*css3标准声明*/
    -moz-box-direction:reverse;
    box-direction:reverse;
}
</style>
</head>
<body>
<div class=d1>左侧布局</div>
<div class=d2>中间布局</div>
<div class=d3>右侧布局</div>
</body>
</html>
```

可以发现此实例代码与上一个实例代码基本相同，只不过多了一个 box-direction 属性设置，此处设置布局进行反向显示。

在 Firefox 62.0 中浏览效果如图 15-10 所示，可以发现与上一个图形相比，左侧布局和右侧布局进行了互换。



图 15-10　盒子布局顺序设置

15.3.3　盒子布局位置 box-ordinal-group

box-ordinal-group 属性设置盒子中每个子元素在盒子中的具体位置。

语法格式如下：

```
box-ordinal-group:<integer>
```

参数值 integer 是一个自然数，从 1 开始，用来设置子元素的位置序号，子元素分别根据这个属性从小到大进行排列。默认情况下，子元素会根据元素的位置进行排列。如果没有知道 box-ordinal-group 属性值的子元素，则其序号默认都为 1，并且序号相同的元素将按照它们在文档中加载的顺序进行排列。

【例 15.10】（实例文件：ch15\15.10.html）

```
<!DOCTYPE html>
<html>
<head>
<title>
box-ordinal-group
</title>
<style>
body{
   margin:0;
   padding:0;
   text-align:center
   background-color:#d9bfe8;
}
.box{
    margin:auto;
    text-align:center;
    width:988px;
    display:-moz-box;
    display:box;
    box-orient:vertical;
    -moz-box-orient:vertical;
}
.box1{
       -moz-box-ordinal-group:2;
       box-ordinal-group:2;
}
.box2{
       -moz-box-ordinal-group:3;
       box-ordinal-group:3;
}
.box3{
       -moz-box-ordinal-group:1;
       box-ordinal-group:1;
}
.box4{
       -moz-box-ordinal-group:4;
       box-ordinal-group:4;
}
```

```
</style>
</head>
<body>
<div class=box>
<div class=box1><img src=1.jpg/></div>
<div class=box2><img src=2.jpg/></div>
<div class=box3><img src=3.jpg/></div>
<div class=box4><img src=4.jpg/></div>
</div>
</body>
</html>
```

在上面的代码中，类选择器 box 中代码 display:box 设置了容器以盒子方向显示，box-orient:vertical 代码设置排列方向从上到下。在下面的 box1、box2、box3 和 box4 类选择器中通过 box-ordinal-group 属性都设置了显示顺序。

在 Firefox 62.0 中浏览效果如图 15-11 所示，可以看到第 3 个层次显示在第一个和第二个层次之上。



图 15-11　设置层显示顺序

15.3.4　盒子弹性空间 box-flex

box-flex 属性能够灵活地控制子元素在盒子中的显示空间。显示空间包括子元素的宽度和高度，而不只是子元素所在栏目的宽度，也可以说是子元素在盒子中所占的面积。

语法格式如下：

```
box-flex:<number>
```

<number>属性值是一个整数或小数。当盒子中包含多个定义了 box-flex 属性的子元素时，浏览器将会把这些子元素的 box-flex 属性值相加，然后根据它们各自的值占总值的比例来分配盒子剩余的空间。



box-flex 属性只有在盒子拥有确定的空间大小时才能够正确解析，即为盒子定义具体的 width 和 height 属性值。

【例 15.11】（实例文件：ch15\15.11.html）

```
<!DOCTYPE html>
<html>
<head>
<title>
box-flex
</title>
<style>
body{
margin:0;
padding:0;
text-align:center;
}
.box{
height:50px;
text-align:center;
width:960px;
overflow:hidden;
display:box;/*标准声明，盒子显示*/
display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
orient:horizontal;/*定义元素为盒子显示*/
-mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
box-orient:horizontal;/*css3标准声明*/
}
.d1{
background-color:#F6F;
width:180px;
height:500px;
}
.d2,.d3{
border:solid 1px #CCC;
margin:2px;
}
.d2{
-moz-box-flex:2;
box-flex:2;
background-color:#3F9;
height:500px;
}
.d3{
-moz-box-flex:4;
box-flex:4;
background-color:#FCd;
height:500px;
}
.d2 div,.d3 div{display:inline;}
</style>
</head>
```

```
<body>
<div class=box>
<div class=d1>左侧布局</div>
<div class=d2>中间布局</div>
<div class=d3>右侧布局</div>
</div>
</body>
</html>
```

上面 CSS 样式代码中，使用 display:box 语句设置容器内元素以盒子方式布局；box-orient:horizontal 语句设置盒子之间在水平方向上并列显示；类选择器 d1 中使用 width 和 height 设置显示层的大小，而在 d2 和 d3 中，使用 box-flex 分别设置两个盒子显示面积。

在 Firefox 62.0 中浏览效果如图 15-12 所示，可以看到右侧布局所占空间比中间布局大。

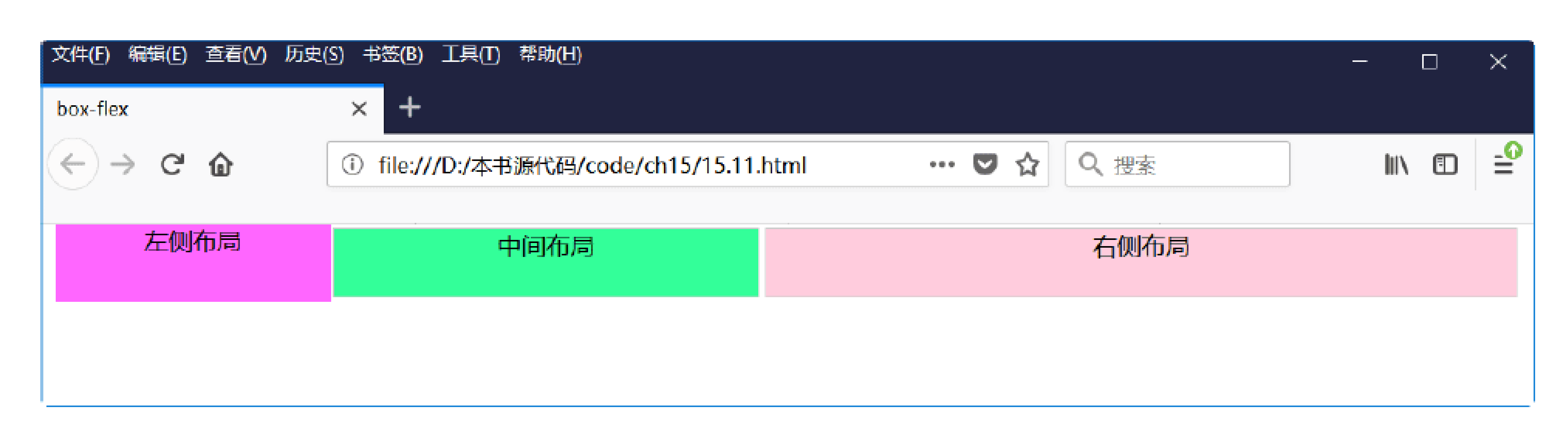


图 15-12　设置盒子面积

15.3.5　管理盒子空间 box-pack 和 box-align

当弹性元素和非弹性元素混合排版时，可能会出现所有子元素的尺寸大于或小于盒子的尺寸，从而出现盒子空间不足或富余的情况，这时就需要一种方法来管理盒子的空间。如果子元素的总尺寸小于盒子的尺寸，则可以使用 box-aling 和 box-pack 属性进行管理。

box-pack 属性可以用于设置子容器在水平轴上的空间分配方式。语法格式如下：

```
box-pack:start|end|center|justify
```

其参数值含义如表 15-6 所示。

15-6　box-pack 属性值

属性值	说明
start	所有子容器都分布在父容器的左侧，右侧留空
end	所有子容器都分布在父容器的右侧，左侧留空
justify	所有子容器平均分布（默认值）
center	平均分配父容器剩余的空间（能压缩子容器的大小，并且有全局居中的效果）

box-align 属性用于管理子容器在竖轴上的空间分配方式。语法格式如下：

```
box-align: start|end|center|baseline|stretch
```

其参数值含义如表 15-7 所示。

表 15-7　box-align 属性值

属性值	说明
start	子容器从父容器顶部开始排列，富余空间显示在盒子底部
end	子容器从父容器底部开始排列，富余空间显示在盒子顶部
center	子容器横向居中，富余空间在子容器两侧分配，上面一半下面一半
baseline	所有盒子沿着它们的基线排列，富余的空间可前可后显示
stretch	每个子元素的高度被调整到适合盒子的高度显示，即所有子容器和父容器保持同一高度

【例 15.12】（实例文件：ch15\15.12.html）

```
<!DOCTYPE html>
<html>
<head>
<title>
box-pack
</title>
<style>
body,html{
height:100%;
width:100%;
}
body{
    margin:0;
    padding:0;
    display:box;/*标准声明，盒子显示*/
    display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
    -mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
    box-orient:horizontal;/*css3标准声明*/
    -moz-box-pack:center;
    box-pack:center;
    -moz-box-align:center;
    box-align:center;
    background:#04082b url(a.jpg) no-repeat top center;
}
.box{
    border:solid 1px red;
    padding:4px;
}
</style>
</head>
<body>
```

```
<div class=box>
<img src=yueji.jpg>
</div>
</body>
</html>
```

上面代码中，display:box 语句定义了容器内元素以盒子形式显示；box-orient:horizontal 定义了盒子水平显示；box-pack:center 定义了盒子两侧空间平均分配；box-align:center 定义了盒子上下两侧平均分配，即图片盒子居中显示。

在 Firefox 62.0 中浏览效果如图 15-13 所示，可以看到中间盒子在容器中部显示。

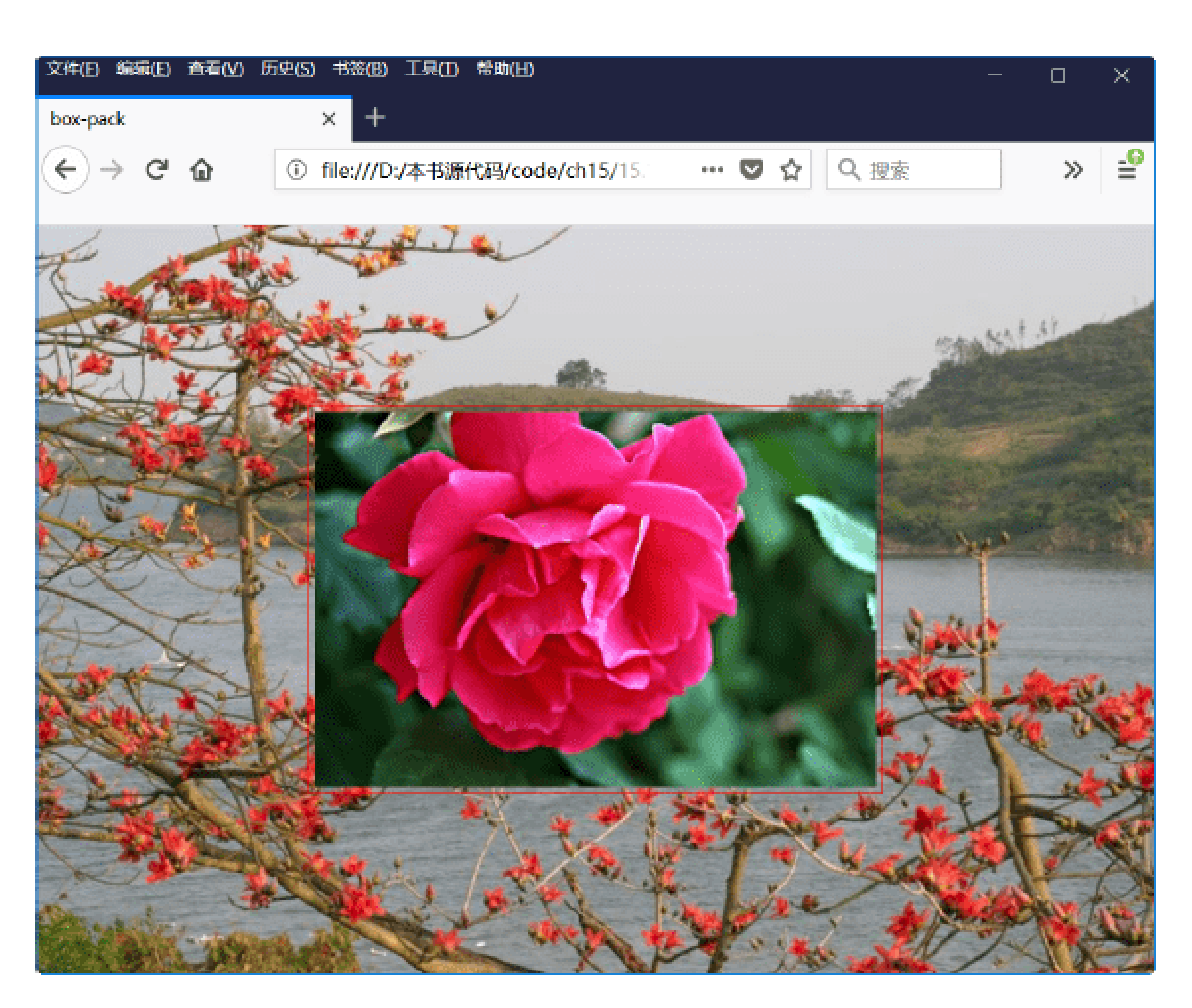


图 15-13　设置盒子中间显示

15.3.6　空间溢出管理 box-lines

弹性布局中盒子内的元素很容易出现空间溢出的现象，与传统的盒子模型一样，CSS3 允许使用 overflow 属性来处理溢出内容的显示。当然，还可以使用 box-lines 属性来避免空间溢出的问题。

语法格式如下：

```
box-lines:single|multiple
```

其中参数值 single 表示子元素都单行或单列显示；multiple 表示子元素可以多行或多列显示。

【例 15.13】（实例文件：ch15\15.13.html）

```
<!DOCTYPE html>
<html>
<head>
<title>
box-lines
</title>
```

```
<style>
.box{
border:solid 1px red;
width:600px;
height:400px;
display:box;/*标准声明，盒子显示*/
display:-moz-box;/*兼容Mozilla Gecko引擎浏览器*/
-mozbox-box-orient:horizontal;/*兼容Mozilla Gecko引擎浏览器*/
-moz-box-lines:multiple;
box-lines:multiple;
}
.box div{
    margin:4px;
    border:solid 1px #aaa;
    -moz-box-flex:1;
    box-flex:1;
}
.box div img{width120px;}
</style>
</head>
<body>
<div class=box>
<div><img src="b.jpg"></div>
<div><img src="c.jpg"></div>
<div><img src="d.jpg"></div>
<div><img src="e.jpg"></div>
<div><img src="f.jpg"></div>
</div>
</body>
```

在 Firefox 62.0 中浏览效果如图 15-14 所示，可以看到右边盒子还是发生溢出现象。这是因为目前各大主流浏览器还没有明确支持这种用法，所有导致 box-lines 属性被实际应用时显示无效。相信在未来的一段时间内，各个浏览器会支持该属性。

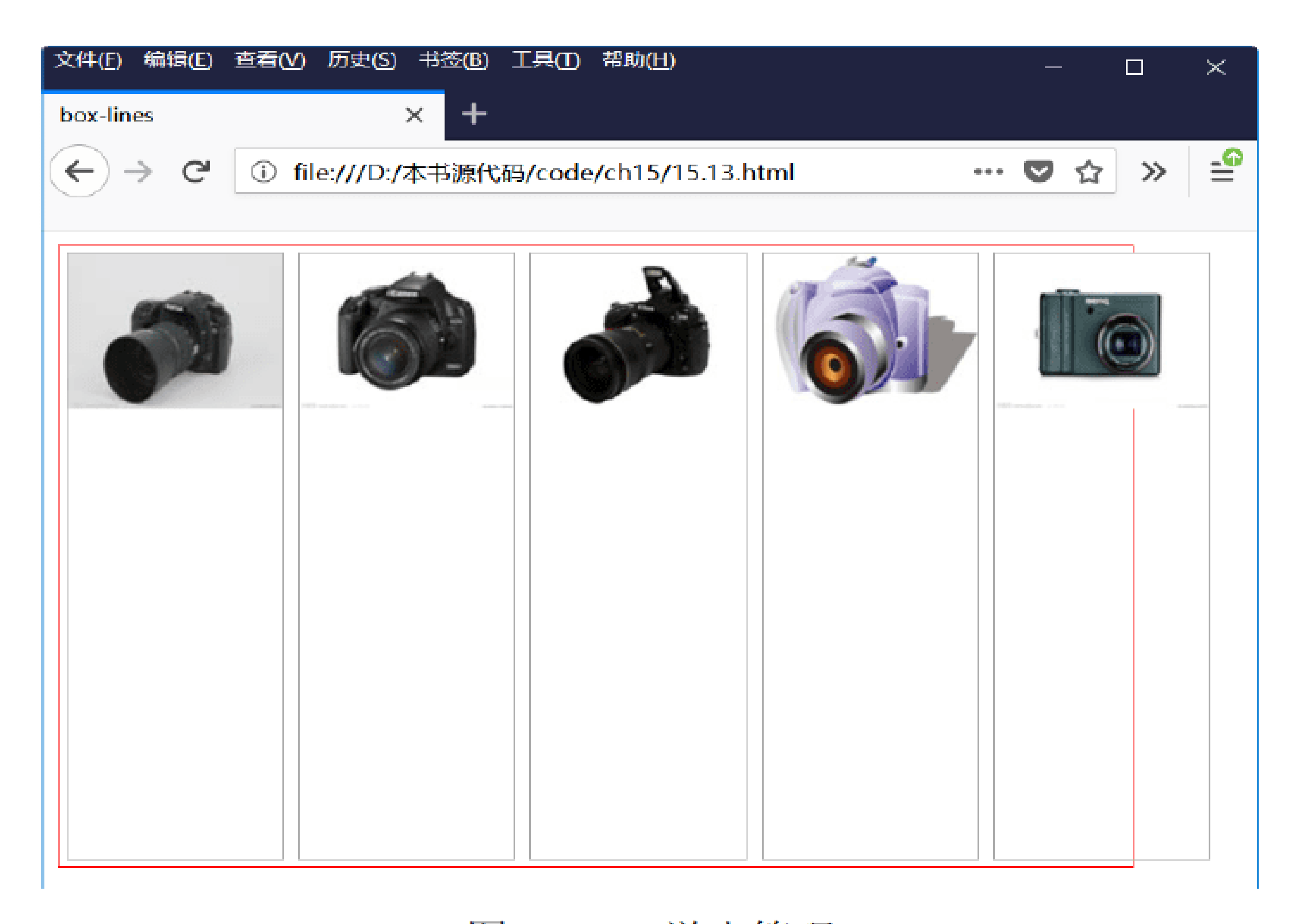


图 15-14　溢出管理

15.4 综合实例 1——图文排版效果

一个宣传页需要包括文字和图片信息。本实例将结合前面学习的盒子模型及其相关属性创建一个旅游宣传页。具体步骤如下：

01 分析需求。

整个宣传页面需要一个 div 层包含并带有边框，div 层包括两个部分：上部空间包含一个图片，下面显示文本信息并带有底边框；下部空间显示两张图片。

02 构建 HTML 页面，使用 DIV 搭建框架。

```
<!DOCTYPE html>
<html>
<head>
<title>图文排版</title>
</head>
<body>
<div class="big">
    <div class="up">
        <img src="top.jpg" border="0" />
            <p>·反季游正流行 众信旅游暑期邀你到南半球过冬 </p>
            <p> ·西安世园会暨旅游推介会今日在沈阳举行！</p>
            <p> ·澳大利亚旅游局中国区首代邓李宝茵八月底卸任</p>
            <p> · “彩虹部落”土族:旅游经济支撑下的文化记忆恢复(组图)</p>
      </div>
      <div class="down">
       <img src="bottom1.jpg" border="0" />    <img
src="bottom2.jpg" border="0" />
      </div>
  </div>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 15-15 所示，可以看到页面自上向下，显示图片、段落信息和图片。

03 添加 CSS 代码，修饰整体 DIV。

```
<style>
*{
    padding:0px;
    margin:0px;
}
body{
    font-family:"宋体";
    font-size:12px;
```

```
    }
    .big{
        width:220px;
        border:#0033FF 1px solid;
        margin:10px 0 0 20px;
    }
    </style>
```

CSS 样式代码在 boyd 标志选择器设置了字体类型和字体大小，并在 big 类选择器中设置整个层的宽带、边框样式和外边距。

在 IE 11.0 中浏览效果如图 15-16 所示，可以看到页面图片信息和文本都在一个矩形盒子内的现象，其边框颜色为蓝色，大小为 1 像素。

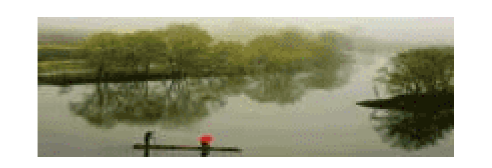





图 15-15　构建 HTML 文档



图 15-16　设置整体 DIV 样式

04 添加 CSS 代码，修饰字体和图片。

```
    .up p{
        margin:5px;
    }
    .up img{
        margin:5px;
        text-align:center;}
    .down{
        text-align:center;
        border-top:#FF0000 1px dashed;
    }
    .down img{
        margin-top:5px;
    }
```

上面代码定义了段落、图片的外边距，如 margin-top:5px 语句设置了下面图片的外边距为 5 像素，两个图片距离是 10 像素。

在 IE 11.0 中浏览效果如图 15-17 所示，可以看到字体居中显示，下面带有一个红色虚线，宽度为 1 像素。



图 15-17　设置各个元素外边距

15.5　综合实例 2——淘宝导购菜单

现在，网上购物已经成为一种时尚，其中淘宝网是网上购物网站影响比较大的网站之一。淘宝网的宣传页面，到处都是。本实例结合前面学习的知识创建一个淘宝网宣传导航页面。具体步骤如下：

01 分析需求。

根据实际效果需要创建一个 div 层。该 div 层包含三部分：一个是左边导航栏；中间图片显示区域；右边是导航栏，然后使用 CSS 颜色设置导航栏字体和边框。

02 构建 HTML 页面，使用 div 搭建框架。

```
<!DOCTYPE html>
<html >
<head>
<title>淘宝网</title>
</head>
<body>
<div class="wrap">
<div class="area">
<div >
<div class="tab area">
  <ul>
    <li class="current"><a href="#">男T恤</a></li>
    <li ><a href="#">男衬衫</a></li>
    <li><a href="#">休闲裤</a></li>
    <li><a href="#">牛仔裤</a></li>
    <li><a href="#">男短裤</a></li>
```

```
       <li><a href="#">西裤</a></li>
       <li><a href="#">皮鞋</a></li>
       <li><a href="#">休闲鞋</a></li>
         <li ><a href="#">男凉鞋</a></li>
     </ul>
   </div>
     <div class="tab area1" >
     <ul>
   <li><a href="#">女T恤</a></li>
       <li><a href="#">女衬衫</a></li>
       <li><a href="#">开衫</a></li>
       <li ><a href="#">女裤</a></li>
       <li><a href="#">女包</a></li>
       <li ><a href="#">男包</a></li>
       <li ><a href="#">皮带</a></li>
       <li><a href="#">登山鞋</a></li>
       <li ><a href="#">户外装</a></li>
     </ul>
   </div>
   </div>
   <div class="img area">
   <img src=nantxu.jpg/>
   </div>
   </div>
   </body>
   </html>
```

在 Firefox 62.0 中浏览效果如图 15-18 所示，三部分内容分别自上而下显示，第一部分是导航菜单栏，第二部分也是一个导航菜单栏，第三部分是一个图片信息。



图 15-18 基本 HTML 显示

03 添加 CSS 代码，修饰整体样式。

```
    <style type="text/css">
    body, p, ul, li{margin:0; padding:0;}
    body{font:12px arial,宋体,sans-serif;}
    .wrap{width:318px;height:248px; background-color:#FFFFFF;
float:left;border: 1px solid #F27B04;}
    .area{width:318px;  float:left;}
    .tab area{width:53px; height:248px;  border-right:1px solid
#F27B04;overflow:hidden; }
    .tab area1{width:53px; height:248px;  border-left:1px solid
#F27B04;overflow:hidden; position:absolute; left:265px; top:1px; }
    .img area{
      width:208px;
      height:248px;
      overflow:hidden;
      position:absolute;
      top:-2px;
      left:55px;
    }
    </style>
```

上面 CSS 样式代码中，设置了 body 页面字体、段落、列表和列表选项的样式。需要注意的是，类选择器 tab_area 定义了左边列表选项，即左边导航菜单，其宽度为 53 像素，高度为 248 像素，边框色为黄色。类选择 tab_area1 定义右边列表选项，即右边导航菜单，其宽度和高度与左侧菜单相同，但在此使用 position 定义了这个 div 层显示的绝对位置，语句为“position:absolute; left:265px; top:1px;”。类选择器 img_area 定义中间图片显示样式，也是使用 position 绝对定位。

在 Firefox 62.0 中浏览效果如图 15-19 所示，可以看到显示了三部分，左右两侧为导航菜单栏，中间是图片。



图 15-19　设置整体布局样式

04 添加 CSS 代码，修饰列表选项。

```
    img{border:0;}
    li{list-style:none;}
    a{font-size:12px; text-decoration:none}
    a:link,a:visited {color:#999;}
    .tab area ul li,.tab area1 ul li
    {width:53px;height:27px;text-align:center;line-height:26px;
float:left;border-bottom:1px solid #F27B04;}
    .tab area ul li a,.tab area1 ul li a{color:#3d3d3d;}
    .tab area ul li.current,.tab area1 ul li.current{ height:27px;
background-color:#F27B04;}
    .tab area ul li.current a,.tab area1 ul li.current a{color:#fff;
font-size:12px; font-weight:400; line-height:27px}
```

上面 CSS 样式代码，完成对字体大小、颜色、是否带有下画线等属性的定义。

在 Firefox 62.0 中浏览效果如图 15-20 所示，可以看到网页中左右两个导航菜单栏，相对于上面图形，字体颜色发生了变化，大小也发生了变化。



图 15-20　修饰列表选项

15.6　专家解惑

1. 如何理解 margin 的加倍问题？

当 div 层被设置为 float 时，在 IE 下设置的 margin 会加倍，这是一个 ie 都存在的 bug。其解决办法是，在这个 div 里面加上 display:inline;。

例如：

```
    <#div id="imfloat"></#DIV>
```

相应的 css 为：

```
    #IamFloat{
    float:left;
```

```
    margin:5px;
    display:inline;
    }
```

2. margin:0 auto 表示什么含义？

margin:0 auto 定义元素向上补白 0 像素，左右为自动使用。这样按照浏览器解析习惯是可以让页面居中显示的，一般这个语句会在 body 标记中。在使用 margin:0 auto 语句使页面居中的时候，一定要给元素一个高度并且不要让元素浮动，即不要加 float，否则效果失效。

第16章　CSS3+DIV页面基本排版

使用CSS+DIV页面排版，已经逐步替代了使用table表格排版，相比较table表格，DIV+CSS页面排版是一种更好的、更有亲和力的、更灵活的，而且功能更强大的网站设计方法。

16.1　CSS3 排版概念

DIV 在 CSS3+DIV 页面排版中是一个块的概念，DIV 的起始标记和结束标记之间的所有内容都是用来构成这个块的，其中所包含元素特性由 DIV 标记属性来控制，或者是通过使用样式表格式化这个块来进行控制。CSS3+DIV 页面排版思想是首先在整体上进行<div>标记的分块，然后对各个块进行 CSS 定位，最后在各个块中添加相应的内容。

16.1.1　将页面用 DIV 分块

使用 DIV+CSS 页面排版布局，需要对网页有一个整体构思，即网页可以划分几个部分。例如上中下结构，还是左右两列结构，还是三列结构。这时就可以根据网页构思，将页面划分几个 DIV 块，用来存放不同的内容。当然了，大块中还可以存放不同的小块。最后，通过 CSS 属性，对这些 DIV 进行定位。

在现在的网页设计中，一般情况下的网站都是上中下结构，即上面是页面头部，中间是页面内容，最下面是页脚，整个上中下结构最后放到一个 DIV 容器中，方便控制。页面头部一般用来存放 Logo 和导航菜单，页面内容包含页面要展示的信息、链接和广告等，页脚存放的是版权信息和联系方式等。

将上中下结构放置到一个 DIV 容器中，方便后面排版并且方便对页面进行整体调整，如图 16-1 所示。

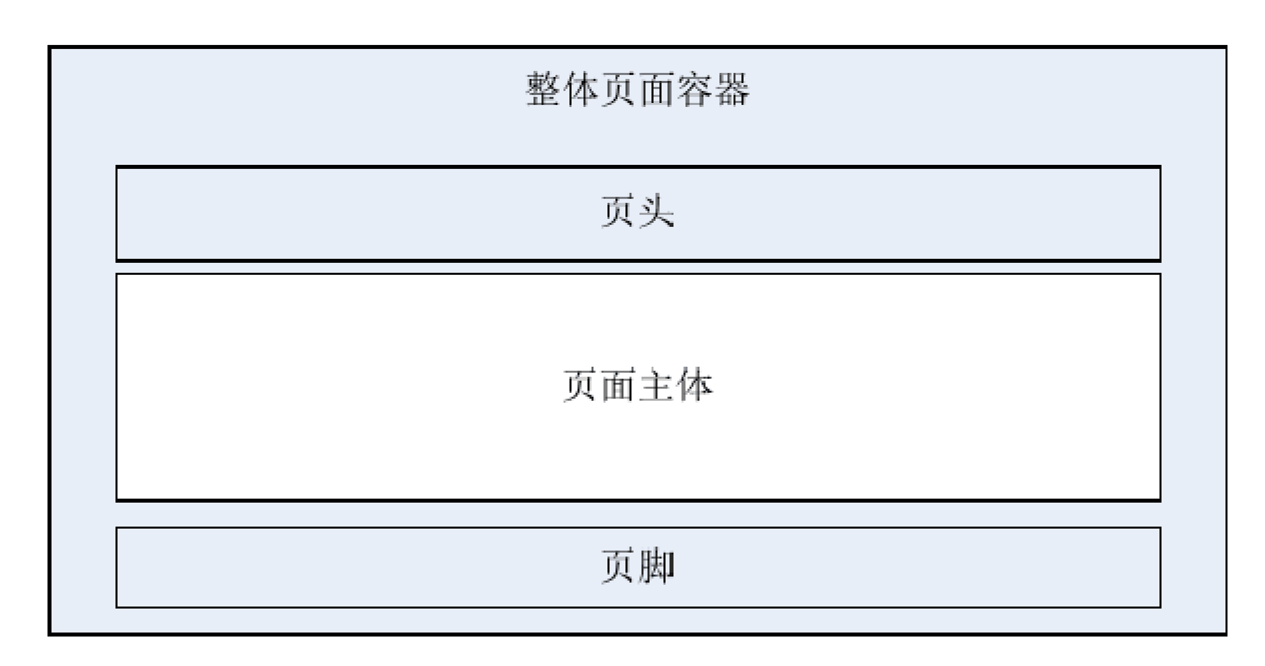


图 16-1　上中下页面结构

16.1.2　设置各块位置

复杂的网页布局，不是单纯的一种结构，而是包含多种网页结构。例如总体上是上中下，中间内分为两列布局等，如图 16-2 所示。

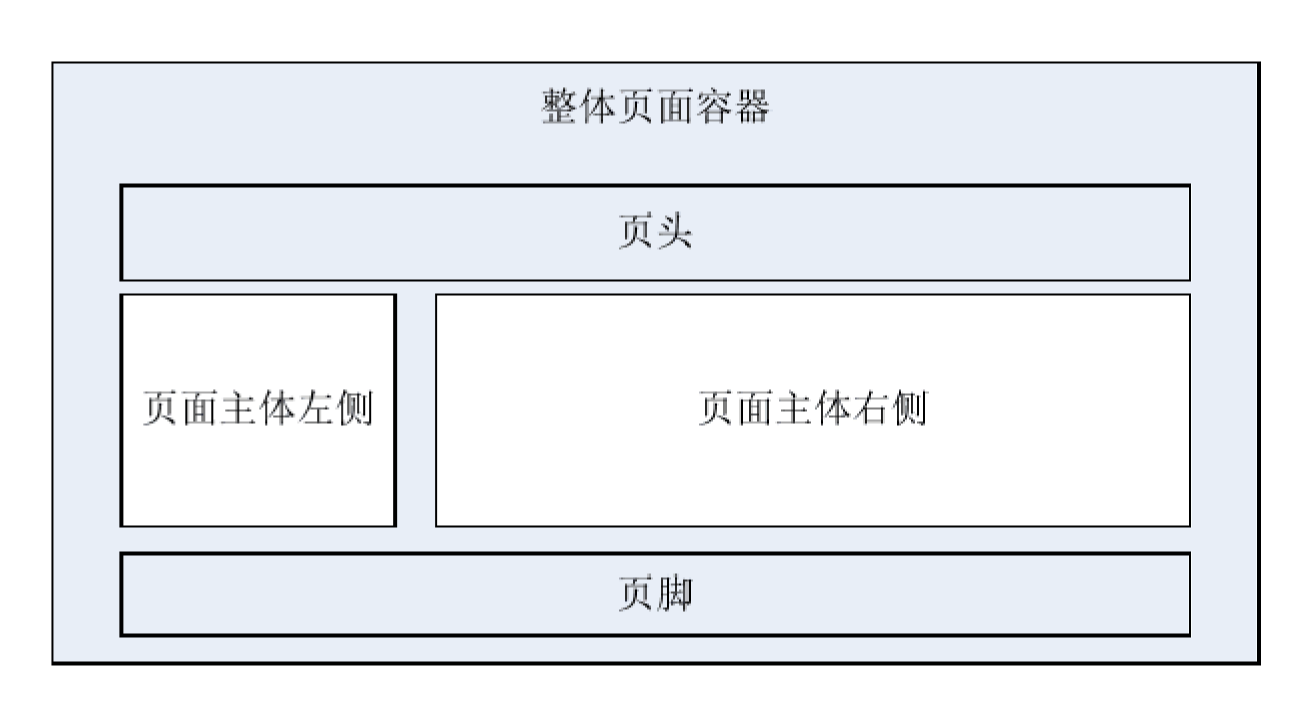


图 16-2　复杂页面布局

页面总体结构确认后，一般情况下，页头和页脚变化就不大了。会发生变化的，就是页面主体，此时需要根据页面展示的内容，决定中间布局采用什么的样式，三列水平分布还是两列分布等。

16.1.3　用 CSS 定位

页面版式确定后，就可以利用 CSS 对 DIV 进行定位，使其在指定位置出现，从而实现对页面的整体规划。然后再向各个页面添加内容。

本节将以图 16-2 为实现蓝图，创建一个总体为上中下布局，页面主体布局为左右布局的页面的 CSS 定位实例。

【例 16.1】（实例文件：ch16\16.1.html）

1. 创建 HTML 页面，使用 DIV 构建层

首先构建 HTML 网页，使用 DIV 划分最基本的布局块。其代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>CSS排版</title><body>
<div id="container">
<div id="banner">页面头部</div>
<div id=content >
<div id="right">
页面主体右侧
</div>
<div id="left">
页面主体左侧
</div>
</div>
<div id="footer">页脚</div>
</div>
</body>
</html>
```

上面代码中，创建了 5 个层，其中 ID 名称为 container 的 div 层，是一个布局容器，即所有的页面结构和内容都是在这个容器内实现；名称为 banner 的 div 层，是页头部分；名称为 footer 的 div 层，是页脚部分。名称为 content 的 div 层，是中间主体，该层包含了两个层，一个是 right 层，一个 left 层，分别放置不同的内容。

在 IE 11.0 中浏览效果如图 16-3 所示，可以看到中显示了这几个层，从上到下一次排列。

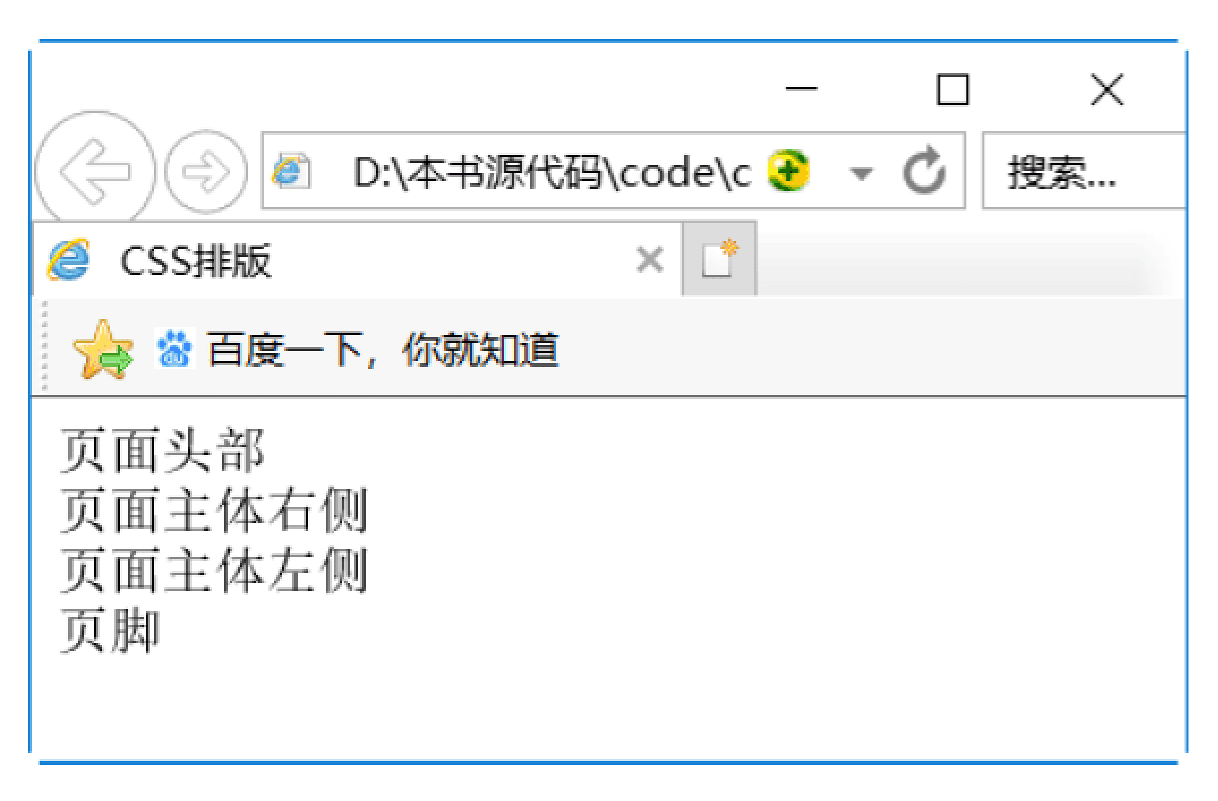


图 16-3　创建 div 层

2. CSS 设置网页整体样式

然后需要对 body 标记和 container 层（布局容器）进行 CSS 修饰，从而对整体样式进行定义。代码如下所示。

```
<style type="text/css">
body {
     margin:0px;
```

```
        font-size:16px;
        font-family:"幼圆";}
    #container{
        position:relative;
        width:100%;}
    </style>
```

上面代码只是设置了文字大小、字体类型、布局容器 container 的宽度、层定位方式，布局容器撑满整个浏览器。

在 IE 11.0 中浏览效果如图 16-4 所示，可以看到此时相比较上一个显示页面，发生的变化不大，只不过字体类型和字体大小发生变化，因为 container 没有带有边框和背景色无法显示该层。

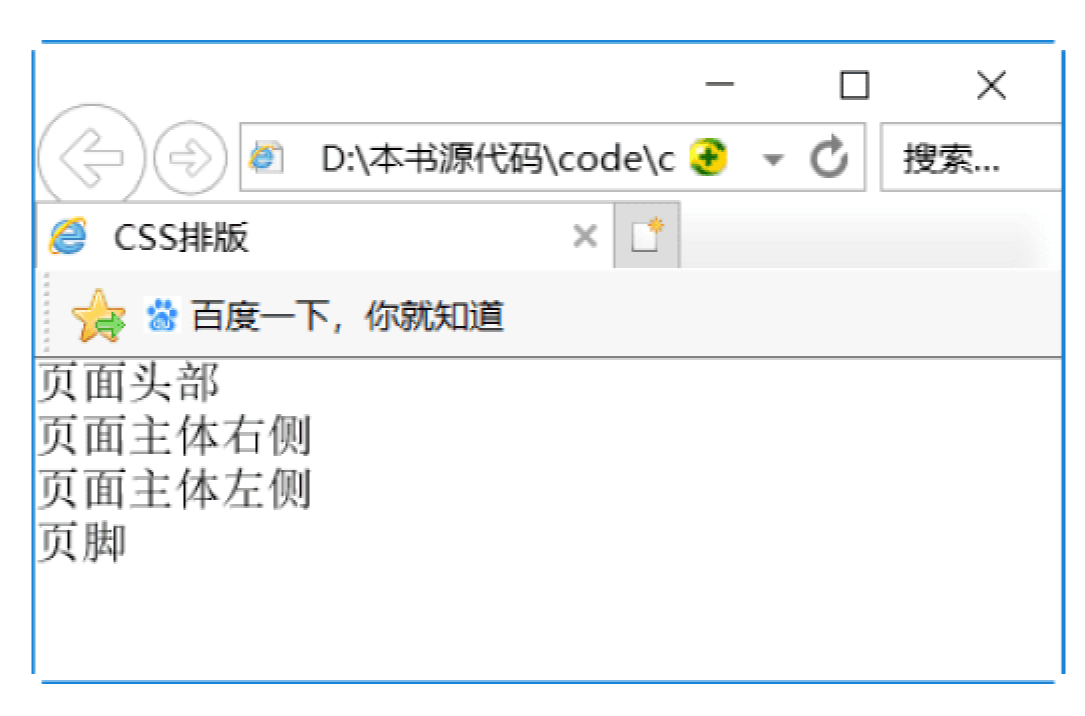


图 16-4　设置全局 CSS 样式

3. CSS 定义页头部分

接下来就可以使用 CSS 对页头进行定位，即 banner 层，使其在网页上显示。代码如下：

```
    #banner{
       height:80px;
       border:1px solid #000000;
       text-align:center;
       background-color:#a2d9ff;
       padding:10px;
       margin-bottom:2px;}
```

上面首先设置了 banner 层的高度为 80 像素，宽度充满整个 container 布局容器，下面分别设置了边框样式、字体对齐方式、背景色、内边距和外边距的底部等。

在 IE 11.0 中浏览效果如图 16-5 所示，可以看到在页面顶部显示了一个浅绿色的边框，边框充满整个浏览器，边框中间显示了一个“页面头部”的文本信息。

4. CSS 定义页面主体

在页面主体如果两个层并列显示，需要使用 float 属性，将一个层设置到左边，一个层设置到右边。其代码如下：

```
#right{
  float:right;
  text-align:center;
  width:80%;
 border:1px solid #ddeecc;
margin-left:1px;
height:200px; }
#left{
    float:left;
    width:19%;
    border:1px solid #000000;
    text-align:center;
height:200px;
background-color:#bcbcbc;}
```

上面代码设置了这两个层的宽带，right 层占有空间的 80%，left 层占有空间的 19%，并分别设置了两个层的边框样式，对齐方式，背景色等。

在 IE 11.0 中浏览效果如图 16-6 所示，可以看到页面主体部分，分为两个层并列显示，左边背景色为灰色，占用空间较小，右侧背景色为白色，占用空间较大。

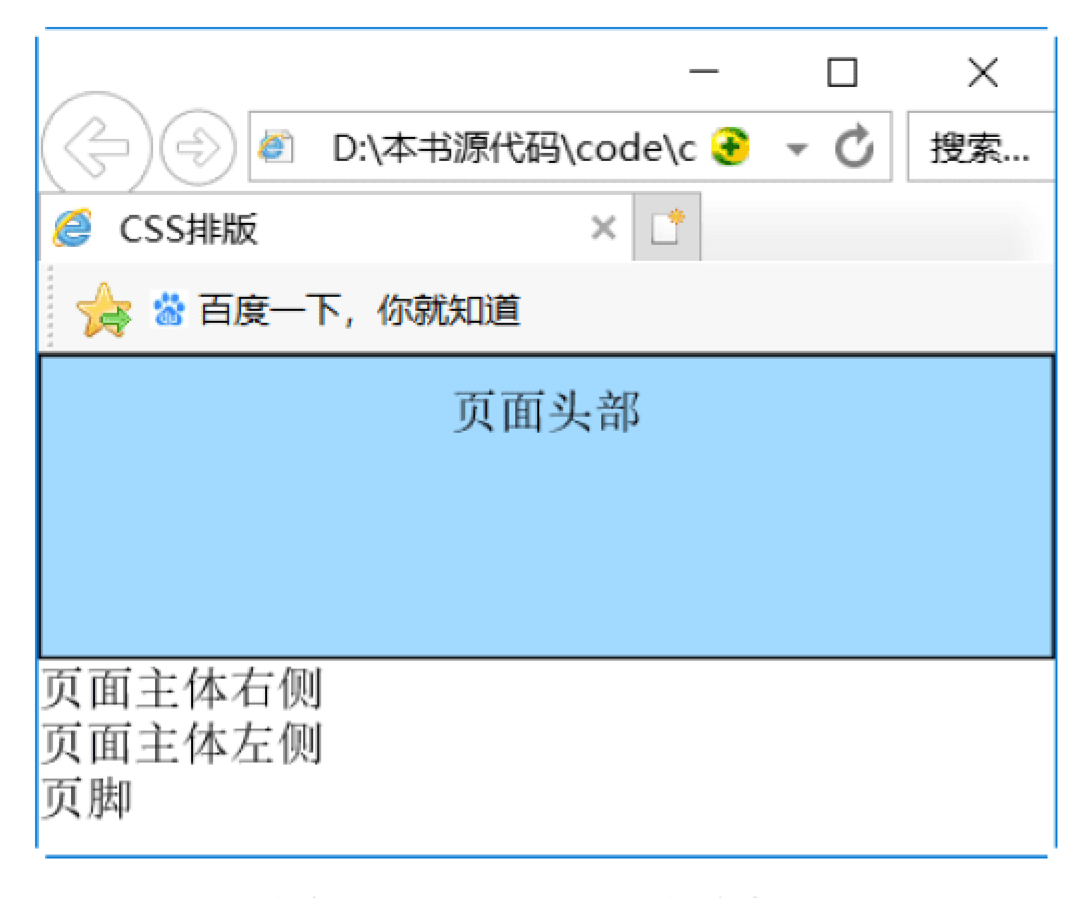


图 16-5　CSS 设置页头

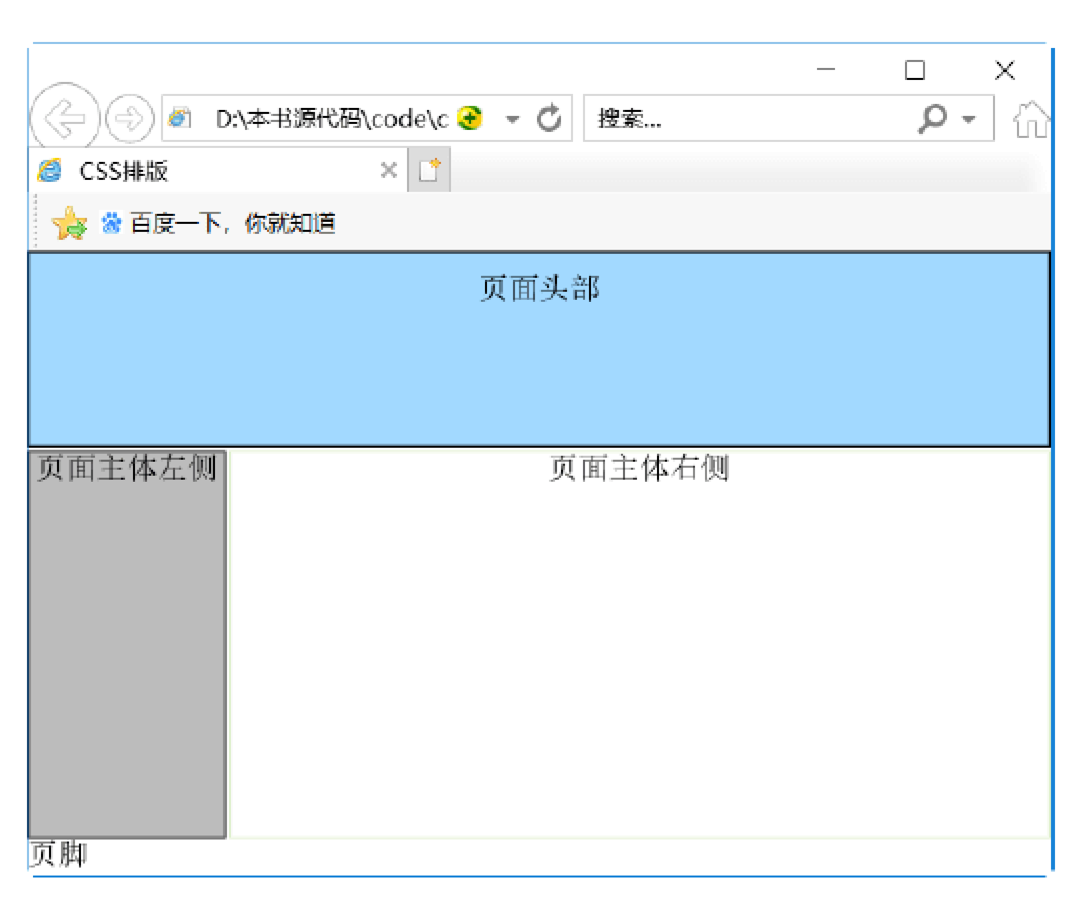


图 16-6　CSS 定位中间主体

5. CSS 定义页脚

最后需要设置页脚部分，页脚一般在主体的下面。因为页面主体中使用了 float 属性设置层浮动，所以需要在页脚层设置 clear 属性，使其不受浮动的影响。代码如下：

```
#footer{
     clear:both;          /* 不受float影响 */
     text-align:center;
     height:30px;
     border:1px solid #000000;
     background-color:#ddeecc;
```

```
}
```

上面代码设置页脚对齐方式、高度、边框和背景色等。在 IE 11.0 中浏览效果如图 16-7 所示，可以看到页面底部显示了一个边框，背景色味浅绿色，边框充满整个 DIV 布局容器。

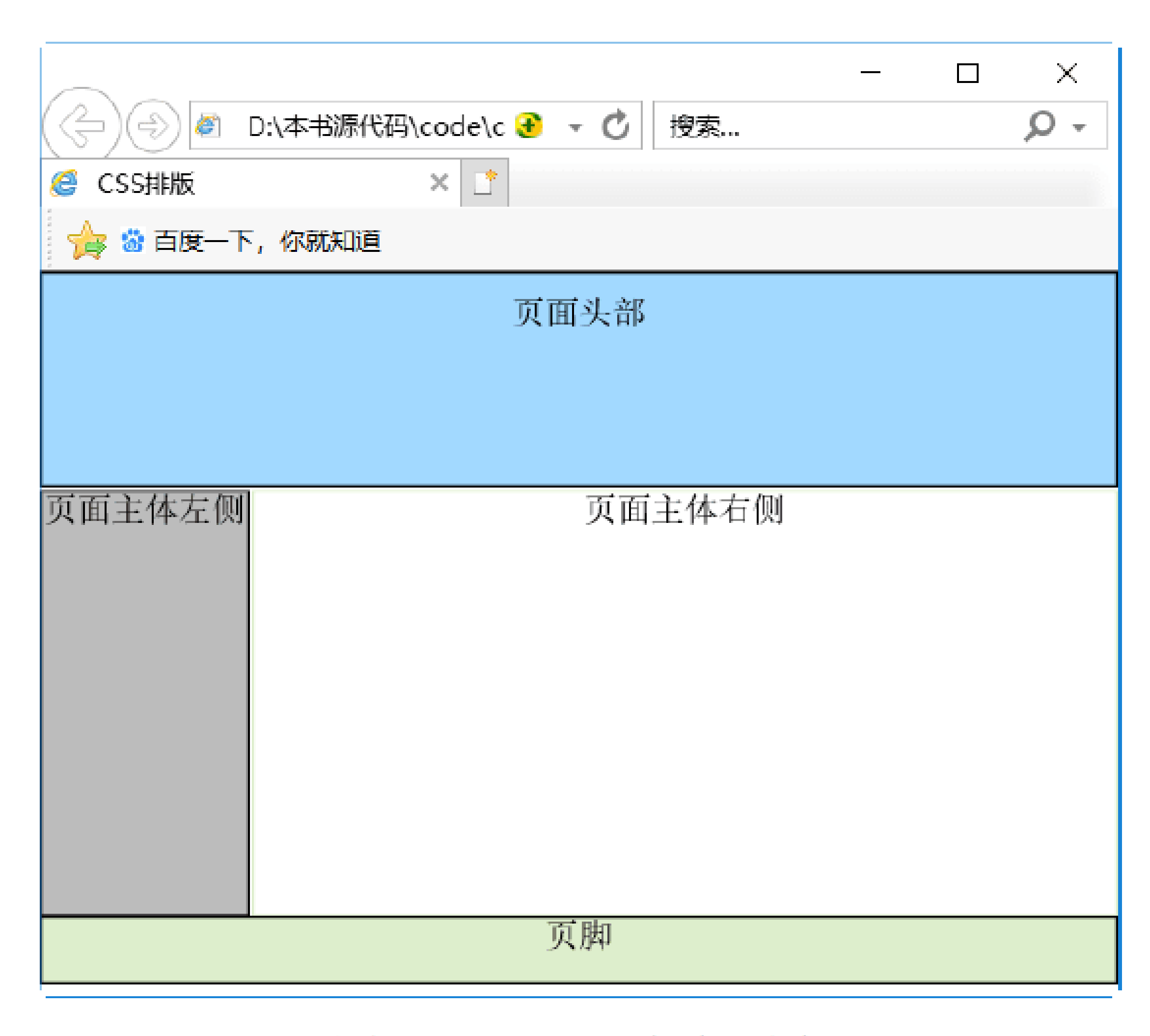


图 16-7　CSS 定义页脚

16.2　固定宽度布局

在进行网页设计时，不得不考虑网页布局宽度，因为浏览者会使用大小不同的浏览器查看页面，设计宽度过大，有的浏览器就不能完全显示出来；设计宽度过小，显示内容就不能完成显示并容易出现滚动条。

16.2.1　上中下版式

网页开发过程中，有几种比较经典的网页排版方式，包括宽度固定的上中下版式、宽度固定的左右版式、自适应宽度布局和浮动布局等。这些版式会经常在网页设计时出现，并且经常被用到各种类型的网站开发中。

宽度固定的上中下版式布局应用非常广泛。但其一般用在简单信息介绍上，而不用在网站首页上。这种版式布局简单，包含信息较少。

【例 16.2】（实例文件：ch16\16.2.html）

1. 创建 HTML 网页，使用 div 层构建块

首先需要使用 DIV 块对页面区域进行划分，使其符合如图 16-8 的页面布局。基本代码如下：

```
    <!DOCTYPE html>
    <html>
    <head>
    <title>上中下排版</title>
    </head>
    <body>
    <div class="big">
    <div class="up">
    <p><a href="#">首页</a><a href="#">环保扫描</a><a href="#">环保科技</a><a
href="#">低碳经济</a><a href="#">土壤绿化</a></p>
    </div>
    <div class="middle">
    <br />
    <h1>拒绝使用一次性用品</h1>
    <p>在现代社会生活中，商品的废弃和任意处理是普遍的，特别是一次性物品使用激增。据统计，
英国人每年抛弃25亿块尿布；……. . </p>
    </div>
    <div class="down"><br />
    <p><a href="#">关于我们</a> | <a href="#">免责声明</a> | <a href="#">联系我
们</a> | <a href="#">生态中国</a> | <a href="#">联系我们</a></p>
    <p>2018 &copy; 世界环保联合会郑州办事处 技术支持</p>
    </div>
    </div>
    </body>
    </html>
```

上面代码创建了 4 个层，层 big 是 DIV 布局容器，用来存放其他的 DIV 块。层 up 表示页头部分，层 middle 表示页面主体，层 down 表示页脚部分。

在 IE 11.0 中浏览效果如图 16-8 所示，可以看到页面显示了三个区域信息，顶部显示的是超级链接部分，中间显示的是段落信息，底部显示的地址和版权信息。



图 16-8　实现 DIV 块

2. 使用 CSS 定义整体

上面页面显示时，字体样式非常丑陋，布局不合理。此时需要使用 CSS 代码，对页面整体样式，进行修饰。代码如下：

```
<style>
*{
padding:0px;
margin:0px;
}
body{
font-family:"幼圆";
font-size:12px;
color:green;
}
.big{
width:900px;
margin:0 auto 0 auto;
}
</style>
```

上面代码定义了页面整体样式，例如字体类型为幼圆，字体大小为 12 像素，字体颜色为绿色，布局容器 big 的宽带为 900 像素。“margin:0 auto 0 auto”语句表示该块与页面的上下边界为 0，左右自动调整。

在 IE 11.0 中浏览效果如图 16-9 所示，可以看到页面字体变小，字体颜色为绿色，并充满整个页面，页面宽度为 900 像素。



图 16-9　定义整体样式

3. 使用 CSS 定义页头部分

下面就可以使用 CSS 定义页头部分，即导航菜单。代码如下：

```
.up{
width:900px;
```

```
height:100px;
background-image:url(17.jpg);
background-repeat:no-repeat;
}
```

在类选择器 up 中，CSS 定义层的宽度和高度，其宽度为 900 像素，并定义了背景图片。

在 IE 11.0 中浏览效果如图 16-10 所示，可以看到页面顶部显示了一个背景图，并且超级链接以一定距离显示，以绝对定位方式在页头显示。



图 16-10　CSS 修饰页头

4. 使用 CSS 定义页面主体

下面需要使用 CSS 定义页面主体，即定义层和段落信息。代码如下：

```
.middle{
border:1px #ddeecc solid;
margin-top:10px;
}
```

在类选择器 middle 中，定义了边框样式，和内边距距离，此处层的宽度和 big 层宽度一致。

在 IE 11.0 中浏览效果如图 16-11 所示，可以看到中间部分以边框形式显示，标题居中显示，段落缩进两个字符显示。



图 16-11　CSS 定义中间部分

5. 使用 CSS 定义页脚部分

定义页脚部分代码如下：

```
.down{
background-color:#CCCCCC;
height:80px;
text-align:center;
}
```

上面代码中，类选择器 down 定义了背景颜色，高度和对齐方式。其他选择器定义超级链接的样式。

在 IE 11.0 中浏览效果如图 16-12 所示，可以看到页面底部显示了一个灰色矩形框，其版权信息和地址信息居中显示。



图 16-12　设置页脚部分

从上面的效果可以看到，这个页面一共分为三个部分，第一部分包含图片和菜单栏，这一部分放到页头，是上中下版式的“上”。第二个部分是中间的内容部分，即页面主体，用于存放要显示的文本信息，是上中下版式的“中”，第三个部分是页面底部，包含地址和版权信息的页脚，是上中下版式的“下”。

16.2.2　左右版式

在页面排版中，有时会根据内容需要页面主体分为左右两个部分显示，用来存放不同的信息内容。实际上上这也是一种宽度固定的版式。

下面实例的框架大体上也是上中下结构，只不过页面主体部分又分了左右版式。

【例 16.3】（实例文件：ch16\16.3.html）

1. 创建 HTML 网页，使用 DIV 构建块

在 HTML 页面，将 DIV 框架和所要显示的内容显示出来，并将要引用的样式名称定义好。

代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>茶网</title>
</head>
<body>
<div id="container">
  <div id="banner">
    <img src="b.jpg" border="0">
  </div>
  <div id="links">
    <ul>
      <li>首页</li>
      <li>茶业动态</li>
      <li>名茶荟萃</li>
      <li>茶与文化</li>
      <li>茶艺茶道</li>
      <li>鉴茶品茶</li>
      <li>茶与健康</li>
     <li>茶语清心</li>
    </ul>    <br />
  </div>
  <div id="leftbar">
      <p class="lefttitle">名人与茶</p>
      <p>.三文鱼茶泡饭</p>
       <p>.董小宛的茶泡饭</p>
       <p>.人生百味一盏茶</p>
      <p>.我家的茶事</p>
    <p class="lefttitle">茶事掌故</p>
     <p>.“峨眉雪芽”的由来</p>
    <p>.茶文化的养生术</p>
     <p>.老北京的花茶</p>
    <p>.古代洗茶的原因和来历</p>
  </div>
  <div id="content">
    <h4>人生茶境</h4>
    <p>“喝茶当于瓦纸窗下，清泉绿茶，用素雅的陶瓷茶具，同二三人共饮，得半日之闲，可抵
十年的尘梦。”</p>
<p>对中国人来说，“茶”是一个温暖的字。.................</p>
  </div>
  <div id="footer">版权所有 2018.08.12</div>
</div>
</body>
</html>
```

上面代码定义了几个层，用来构建页面布局。其中层 container 作为布局容器，banner 作为

页面图形 logo，links 层作为页面导航，leftbar 层作为左侧内容部分，content 层作为右侧内容部分，footer 层作为页脚部分。

在 IE 11.0 中浏览效果如图 16-13 所示，可以看到页面上部显示了一张图片，下面是超级链接、段落信息，最后是地址信息等。

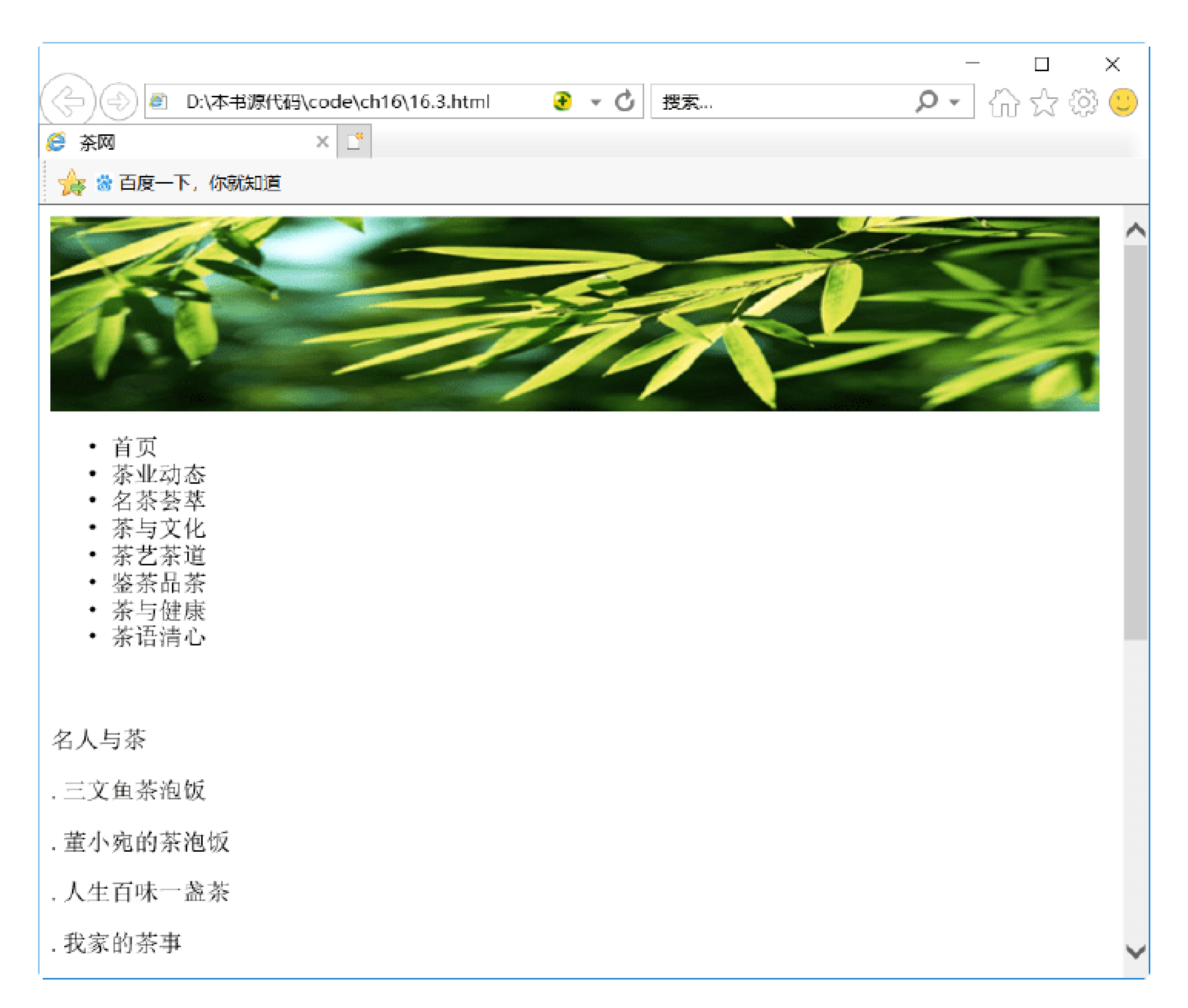


图 16-13 div 层构建布局样式

2. CSS 定义页面整体样式

首先需要定义整体样式，例如网页中字体类型或对齐方式等。代码如下：

```
<style>
body, html{
  margin:0px; padding:0px;
  text-align:center;
}
#container{
  position: relative;
  margin: 0 auto;
  padding:0px;
  width:700px;
  text-align: left;}
</style>
```

上面代码中，类选择器 container 定义了布局容器的定位方式为相对定位，宽度为 700 像素，文本左对齐，内外边距都为 0 像素。

3. CSS 定义页头部分

此网页的页头包含两部分：一个是页面 logo；另一个是页面的导航菜单。定义这两个层 CSS 代码如下：

```
#banner{
  margin:0px; padding:0px;
}
#links{
  font-size:12px;
  margin:-18px 0px 0px 0px;
  padding:0px;
  position:relative;
}
#links ul{
list-style-type:none;
padding:0px; margin:0px;
width:700px;
}
#links ul li{
text-align:center;
width:80px;
display:block;
float:left;
color:#ddeecc;
}
#links br{
display:none;
}
```

上面代码中，ID 选择器 banner 定义了内外边距都是 0 像素，ID 选择器 links 定义了导航菜单的样式，例如字体大小为 12 像素，定位方式为相对定位等。

在 IE 11.0 中浏览效果如图 16-14 所示，可以看到页面导航部分在图像上显示，并且每个菜单相隔一定距离。



图 16-14　定义页头部分

4. CSS 定义页面主体左侧部分

使用 CSS 代码定义页面主体左侧部分。

```
#leftbar{
  background-color:#d2e7ff;
  text-align:center;
  font-size:12px;
  width:150px;
float:left;
  padding-top:0px;
  padding-bottom:30px;
  margin:0px;
}
#leftbar p{
padding-left:12px;
padding-right:12px;
text-align:left;
}
.lefttitle{
background-color:green;
border:1px solid #ddeecc;
}
```

上面选择器 leftbar 中，定义了层背景色、对齐方式、字体大小和左侧 div 层的宽度，这里使用 float 定义层在水平方向上浮动定位。

在 IE 11.0 中浏览效果如图 16-15 所示，可以看到页面左侧部分以矩形框显示，包含了一些简单的页面导航。



图 16-15 CSS 定义页面左侧

5. CSS 定义页面主体右侧部分

使用 CSS 代码定义页面主体右侧部分。

```
#content{
  font-size:12px;
  float:left;
width:550px;
  padding:5px 0px 30px 0px;
  margin:0px;
}
#content p, #content h4{
padding-left:20px;
padding-right:15px;
text-indent:2em;
}
h4{
text-decoration:underline;
color:#0078aa;
padding-top:15px;
font-size:16px;
}
```

代码中 ID 选择器 content，用来定义字体大小、右侧 div 层宽度，内外边距等。在 IE 11.0 中浏览效果如图 16-16 所示，可以看到右侧部分的段落字体变小，段落缩进了两个单元格。



图 16-16　定义右侧内容

6. CSS 定义页脚部分

如果上面的层使用了浮动定位，那么页脚一般需要使用 clear 去掉浮动所带来的影响。代码如下：

```
#footer{
  clear:both;
font-size:12px;
```

```
    width:100%;
    padding:3px 0px 3px 0px;
    text-align:center;
    margin:0px;
    background-color:#b0cfff;
  }
```

footer 选择器中，定义了层的宽度，即充满整个布局容器，字体大小为 12 像素，居中对齐和背景色。在 IE 11.0 中浏览效果如图 16-17 所示，可以看到页脚显示了一个矩形框，背景色为浅蓝色，矩形框内显示了版权信息。



图 16-17　定义页脚

16.3　新增 CSS3 多列布局

在 CSS3 没有出来之前，网页设计者如果要设计多列布局，不外乎有两种方式，一种是浮动布局，另一种是定位布局。浮动布局比较灵活，但容易发生错位，这是需要添加大量的附加代码或无用的换行标记，增加了不必要的工作量。定位布局可以精确的确定位置，不会发生错位，但无法满足模块的适应能力。为了解决多列布局的难题，CSS3 新增了多列自动布局。

16.3.1　列宽度 column-width

在 CSS3 中，可以使用 column-width 属性定义多列布局中每列的宽度，可以单独使用，也可以和其他多列布局属性组合使用。

column-width 语法格式如下：

```
column-width： [<length> | auto]
```

其中属性值<length>是由浮点数和单位标识符组成的长度值，不可为负值。auto 根据浏览器计算值自动设置。

【例 16.4】（实例文件：ch16\16.4.html）

```
<!DOCTYPE html>
<html>
<head>
<title>多列布局属性</title>
<style>
body{
   -moz-column-width:300px;/*兼容Webkit引擎，指定列宽是300像素*/
   column-width:300px; /*CSS3标准化指定列宽是300像素*/
}
h1{
   color:#333333;
   background-color:#DCDCDC;
   padding:5px 8px;
   font-size:20px;
   text-align:center;
    padding:12px;
}
h2{
   font-size:16px;text-align:center;
}
p{color:#333333;font-size:14px;line-height:180%;text-indent:2em;}
</style>
</head>
<body>
<h1>春</h1>
<h2>朱自清</h2>
<p>盼望着，盼望着，东风来了，春天的脚步近了。</p>
<p>一切都像刚睡醒的样子，欣欣然张开了眼。山朗润起来了，水涨起来了，太阳的脸红起来了。
</p>
<p>小草偷偷地从土里钻出来，嫩嫩的，绿绿的。园子里，田野里，瞧去，一大片一大片满是的。
坐着，躺着，打两个滚，踢几脚球，赛几趟跑，捉几回迷藏。风轻悄悄的，草软绵绵的。</p>….
</body>
</html>
```

在上面代码 body 标记选择器中，使用 column-width 指定了要显示的多列布局的每列的宽带。下面分别定义标题 h1、h3 和段落 p 的样式，例如字体大小、字体颜色、行高和对齐方式等。

在 IE 11.0 中浏览效果如图 16-18 所示，可以看到页面文章分为两列显示，列宽相同。



图 16-18　设置列宽

16.3.2　列数 column-count

在 CSS3 中，可以直接使用 column-count 指定多列布局的列数，而不需要通过列宽度自动调整列数。

column-count 语法格式如下：

```
column-count: auto | <integer>
```

上面属性值<integer>表示值是一个整数，用于定义栏目的列数，取值为大于 0 的整数。不可以为负值。auto 属性值表示根据浏览器计算值自动设置。

【例 16.5】（实例文件：ch16\16.5.html）

```
<!DOCTYPE html>
<html>
<head>
<title>多列布局属性</title>
<style>
body{
   -moz-column-count:3;/*Webkit引擎定义多列布局列数*/
   column-count:3; /*CSS3标准定义多列布局列数*/
}
h1{
    color:#333333;
     background-color:#DCDCDC;
```

```
        padding:5px 8px;
        font-size:20px;
        text-align:center;
         padding:12px;
    }
    h2{
       font-size:16px;text-align:center;
    }
    p{color:#333333;font-size:14px;line-height:180%;text-indent:2em;}
    </style>
    </head>
    <body>
    <h1>春</h1>
    <h2>朱自清</h2>
    <p>盼望着，盼望着，东风来了，春天的脚步近了。</p>
    <p>一切都像刚睡醒的样子，欣欣然张开了眼。山朗润起来了，水涨起来了，太阳的脸红起来了。
</p>
    <p>小草偷偷地从土里钻出来，嫩嫩的，绿绿的。园子里，田野里，瞧去，一大片一大片满是的。
坐着，躺着，打两个滚，踢几脚球，赛几趟跑，捉几回迷藏。风轻悄悄的，草软绵绵的。
    </p>……..
    </body>
    </html>
```

上面的 CSS 代码除了 column-count 属性设置外，其他样式属性和上一个例子基本相同，就不介绍了。

在 IE 11.0 中浏览效果如图 16-19 所示，可以看到页面根据指定的情况，显示了三列布局，其布局宽度由浏览器自动调整。



图 16-19 设置列数

16.3.3 列间距 column-gap

多列布局中，可以根据内容和喜好的不同，调整多列布局中列之间的距离，从而完成整体

版式规划。在 CSS3 中，column-gap 属性用于定义两列之间的间距。

column-gap 语法格式如下：

```
column-gap: normal | <length>
```

其中属性值 normal 表示根据浏览器默认设置进行解析，一般为 1em；属性值<length>表示值由浮点数和单位标识符组成的长度值，不可为负值。

【例 16.6】（实例文件：ch16\16.6.html）

```
<!DOCTYPE html>
<html>
<head>
<title>多列布局属性</title>
<style>
body{
   -moz-column-count:3; /*Webkit引擎定义多列布局列数*/
   column-count:3; /*CSS3定义多列布局列数*/
   -moz-column-gap:3em; /*Webkit引擎定义多列布局列间距*/
   column-gap:3em; /*CSS3定义多列布局列间距*/
   line-height:2.5em;
}
h1{
   color:#333333;
   background-color:#DCDCDC;
   padding:5px 8px;
   font-size:20px;
   text-align:center;
   padding:12px;
}
h2{
   font-size:16px;text-align:center;
}
p{color:#333333;font-size:14px;line-height:180%;text-indent:2em;}
</style>
</head>
<body>
<h1>春</h1>
<h2>朱自清</h2>
<p>盼望着，盼望着，东风来了，春天的脚步近了。</p>
<p>一切都像刚睡醒的样子，欣欣然张开了眼。山朗润起来了，水涨起来了，太阳的脸红起来了。
</p>
<p>小草偷偷地从土里钻出来，嫩嫩的，绿绿的。园子里，田野里，瞧去，一大片一大片满是的。
坐着，躺着，打两个滚，踢几脚球，赛几趟跑，捉几回迷藏。风轻悄悄的，草软绵绵的。</p>
………
</body>
</html>
```

上面代码中，使用-moz-column-count 私有属性设置了多列布局的列数，-moz-column-gap 私有属性设置列间距为 3em，行高为 2.5em。

在 IE 11.0 中浏览效果如图 16-20 所示，可以看到页面还是分为三个列，但列之间的距离相比原来增大了不少。



图 16-20　设置列间距

16.3.4　列边框样式 column-rule

边框是样式属性中不可缺少的属性之一，通过边框可以容易的界定边框内容，从而划分不同的区域。在多列布局中，同样可以设置多列布局的边框，用于区分不同的列数。在 CSS3 中，边框样式使用 coumn-rule 属性定义，包括边框宽度、边框颜色和边框样式等。

coumn-rule 语法格式如下：

```
column-rule: <length> | <style> | <color>
```

其中属性值含义如表 16-1 所示。

表 16-1　column-rule 属性值

属性值	说明
<length>	由浮点数和单位标识符组成的长度值，不可为负值。用于定义边框宽度，其功能和 column-rule-width 属性相同
<style>	定义边框样式，其功能和 column-rule-style 属性相同
<color>	定义边框颜色，功能和 column-rule-color 属性相同

为了方便网页设计是设计灵活的边框，CSS3 在 column-rule 属性的基础上派生了三个列边框属性，分别为 column-rule-width、column-rule-style 和 column-rule-color 属性值。

column-rule-color 属性用于定义边框颜色，其颜色值接受 CSS3 支持的所有颜色值。column-rule-width 属性定义边框宽度，其属性值可以是任意浮点数。column-rule-style 定义边框样式，其属性值和 border-style 属性值相同，即包括 none、hidden、dotted、dashed、solid、double、groove、ridge、inset 和 ouset。

【例 16.7】（实例文件：ch16\16.7.html）

```
<!DOCTYPE html>
<html>
<head>
<title>多列布局属性</title>
<style>
body{
   -moz-column-count:3;
   column-count:3;
   -moz-column-gap:3em;
   column-gap:3em;
   line-height:2.5em;
   -moz-column-rule:dashed 2px gray;/*Webkit引擎定义多列布局边框样式*/
   column-rule:dashed 2px gray; /*CSS3定义多列布局边框样式*/
}
h1{
    color:#333333;
     background-color:#DCDCDC;
     padding:5px 8px;
     font-size:20px;
     text-align:center;
      padding:12px;
}
h2{
   font-size:16px;text-align:center;
}
p{color:#333333;font-size:14px;line-height:180%;text-indent:2em;}
</style>
</head>
<body>
<h1>春</h1>
<h2>朱自清</h2>
<p>盼望着，盼望着，东风来了，春天的脚步近了。</p>
<p>一切都像刚睡醒的样子，欣欣然张开了眼。山朗润起来了，水涨起来了，太阳的脸红起来了。
</p>
<p>小草偷偷地从土里钻出来，嫩嫩的，绿绿的。园子里，田野里，瞧去，一大片一大片满是的。
坐着，躺着，打两个滚，踢几脚球，赛几趟跑，捉几回迷藏。风轻悄悄的，草软绵绵的。</p>
……
</body>
</html>
```

在 body 标记选择器中，定义了多列布局的列数、列间距和列边框样式，其边框样式是灰色破折线样式，宽度为 2 像素。

在 IE 11.0 中浏览效果如图 16-21 所示，可以看到页面列之间添加了一个边框，其样式为破折线。



图 16-21　定义边框样式

16.4　综合实例——电子企业首页

小型企业网站往往非常简单，导航信息不断但要求突出主题。本实例模拟一个电子企业首页，总体上是上中下结构，中间采取左右版式。具体步骤如下：

01 分析需求。

创建一个上中下的布局，再加上中间的左右版式，需要 5 个 div 层。

02 创建 HTML 网页，使用 div 层分块。

```
<!DOCTYPE html>
<html>
<head>
<title>两列布局</title>
</head>
<body>
<div class="big">
    <div class="nav">
      <ul>
          <li><a href="#">首页</a></li>
            <li><a href="#">企业新闻</a></li>
            <li><a href="#">视听新闻</a></li>
            <li><a href="#">市场快讯</a></li>
             <li><a href="#">投资导引</a></li>
              <li><a href="#">每天报道</a></li>
```

```
            </ul>
         </div>
         <div class="main">
           <div class="left">
             <div class="wen">
             <h1>中国电子首次入选《财富》世界500强 </h1>
             <p>7月7日，美国《财富》杂志发布2011年度世界500强企业排名，中国电子信息产业集
团有限公司首次入选世界500强企业，……</p>
             </div>
           </div>
             <div class="right">
                 <img src="b0.jpg" border="0" />
                 <p>《精算商业管理系统》针对中小企业的特点，集进销存财务管理一体化。帮助企
业全面管理商品账、资金账、往来账、费用账，……</p>
             </div>
         </div>
         <div class="foot">
           <p>2018 新奇工作室 版权所有</p>
         </div>
     </div>
     </body>
     </html>
```

上面代码中，层 big 是一个布局容器，用来存放其他 div 层；nav 层用来作为页头部分，main 层是网页主体，包含两个 div 层，left 层为页面主体左侧，right 层为页面主体右侧。foot 层表示页脚部分。

在 IE 11.0 中浏览效果如图 16-22 所示，可以看到自上而下排列，上面是超级链接信息，下面是文本和图片。



图 16-22 基本 HTML 页面

03 添加 CSS 修饰整体样式。

```
*{
  padding:0px;
  margin:0px;
}
body{
font-family:"宋体";
font-size:12px;
}
.big{
width:900px;
margin:0 auto 0 auto;
}
```

上面代码定义了全局样式，例如网页上字体类型为宋体，字体大小为 12 像素，内外边距都是 0 像素。在 big 选择器中，定义整个布局容器宽度为 900 像素。

在 IE 11.0 中浏览效果如图 16-23 所示，可以看到页面字体变小，并以宋体显示。



图 16-23　定义全局样式

04 添加 CSS，修饰页头部分。

```
.nav{
background-color:#C2E3E9;
width:800px;
height:80px;
background-image:url(80.jpg);
}
.nav ul{
list-style-type:none;
}
.nav li{
float:left;
```

```
}
.nav a{
display:block;
width:120px;
height:20px;
line-height:20px;
text-align:center;
color:#6600CC;
text-decoration:none;
}
.nav a:hover{
width:120px;
height:20px;
color:#990000;
background-color:#CCCCCC;
}
```

在选择器 nav 中，定义了背景色、背景图片、宽度和高度。下面的选择器分别定义了导航菜单的显示样式，例如无序列表不显示符号、字体大小、颜色、对齐方式和行高等。

在 IE 11.0 中浏览效果如图 16-24 所示，可以看到页面顶部显示了一个图片，其导航菜单在图片顶部显示。



图 16-24　定义页头部分

05 添加 CSS，修饰页面主体。

```
.main{
    width:800px;
    height:400px;
}
.left{
width:618px;
float:left;
height:400px;
text-align:center;
border:1px solid #cbcbcb;
}
```

```
.left img{
margin:10px 0 10px 0;
}
.left p{
text-indent:2em;
text-align:left;
font-size:13px;
line-height:30px;
}
.right{
width:178px;
float:left;
height:400px;
text-align:center;
border:1px solid #cbcbcb;
}
.right p{
text-indent:2em;
font-size:13px;
}
```

上面代码中，main 选择器定义中间容器整体宽度为 800 像素，高度为 400 像素；left 选择定义网页主体左侧部分样式，如宽度为 618 像素，高度为 400 像素，float 指定 div 层在左侧浮动布局。Right 选择器定义页面主体右侧部分，例如宽度为 178 像素，高度为 400 像素，文本居中对齐等。

在 IE 11.0 中浏览效果如图 16-25 所示，可以看到页面中间部分分为两个边框显示，左侧部分显示了段落信息，右侧部分显示了图片和文本信息。



图 16-25　定义主体部分

06 添加 CSS，修饰页脚部分。

```
.foot{
height:15px;
text-align:center;
}
.foot p{
font-style:italic;
font-size:10px;
width:800px;
text-align:center;
}
```

上面代码定义了页脚部分显示样式，例如页脚高度为 15 像素，文本居中对齐，段落样式中字体大小为 10 像素，居中对齐等。

在 IE 11.0 中浏览效果如图 16-26 所示，可以看到页脚部分居中显示，其字体以斜体显示。



图 16-26　定义页脚部分

16.5　专家解惑

1. IE 浏览器和 Firefox 浏览器，显示 float 浮动布局会出现不同的效果，为什么？

两个相连的 IDV 块，如果一个设置为左浮动，一个设置为右浮动，这时在 Firefox 浏览器中就会出现设置失效的问题。其原因是 IE 浏览器会根据设置来判断 float 浮动，而在 Firefox

中，如果上一个 float 没有被清除的话，则下一个 float 会自动沿用上一个 float 的设置，而不使用自己的 float 设置。

这个问题的解决办法就是，在每一个 DIV 块设置 float 后，在最后加入一句清除浮动的代码 clear:both，这样就会清除前一个浮动的设置了，下一个 float 也就不会再使用上一个浮动设置，从而使用自己所设置的浮动了。

2. div 层高度设置好，还是不设置好？

在 IE 浏览器中，如果设置了高度值，但是内容很多，会超出所设置的高度，这时浏览器就会自己撑开高度，以达到显示全部内容的效果，不受所设置的高度值限制。而在 Firefox 浏览器中，如果固定了高度的值，那么容器的高度就会被固定住，就算内容过多，他也不会撑开，也会显示全部内容，但是如果容器下面还有内容的话，那么这一块就会与下一块内容重合。

这个问题的解决办法就是，不要设置高度的值，这样浏览器就会根据内容自动判断高度，也不会出现内容重合的问题。

第17章　熟悉jQuery Mobile

针对不同移动设备上显示界面统一的问题，jQuery 又推出了新的函数库 jQuery Mobile。本章节将重点学习 jQuery Mobile 的基础知识。

17.1　认识 jQuery Mobile

jQuery Mobile 是 jQuery 在手机上和平板设备上的版本。jQuery Mobile 不仅会给主流移动平台带来 jQuery 核心库，而且会发布一个完整统一的 jQuery 移动 UI 框架。通过 jQuery Mobile 制作出来的网页能够支持全球主流的移动平台，而且在浏览网页时，能够拥有操作应用软件一样的触碰和滑动效果。

jQuery Mobile 的优势如下：

（1）简单易用：jQuery Mobile 简单易用。页面开发主要使用标记，无须或仅需很少 JavaScript。jQuery Mobile 通过 HTML5 标记和 CSS3 规范来配置和美化页面，对于已经熟悉 HTML5 和 CSS3 的读者来说，上手非常容易，架构清晰。

（2）跨平台：目前大部分的移动设备浏览器都支持 HTML5 标准和 jQuery Mobile，所以可以实现跨不同的移动设备。例如 Android、Apple IOS、BlackBerry、Windows Phone、Symbian 和 MeeGo 等。

（3）提供丰富的函数库：常见的键盘、触碰功能等，开发人员不用编写代码，只需要经过简单的设置，就可以实现需要的功能，大大减少了程序员开发的时间。

（4）丰富的布景主题和 ThemeRoller 工具：jQuery Mobile 提供了布局主题，通过这些主题，可以轻轻松松地快速创建绚丽多彩的网页。通过使用 jQuery UT 的 ThemeRoller 在线工具，只需要在下拉菜单中进行简单的设置，就可以制作出丰富多彩的网页风格，并且可以将代码下载下来应用。

jQuery Mobile 的操作流程如下：

（1）创建 HTML5 文件。

（2）载入 jQuery、jQuery Mobile 和 jQuery Mobile CSS 链接库。

（3）使用 jQuery Mobile 定义的 HTML 标准，编写网页架构和内容。

17.2 跨平台移动设备网页 jQuery Mobile

学习移动设备的网页设计开发，遇到最大的难题是跨浏览器支持的问题。为了解决这个问题，jQuery 推出了新的函数库 jQuery Mobile，主要用于统一当前移动设备的用户界面。

17.2.1 移动设备模拟器

网页制作完成后，需要在移动设备上预览最终的效果。为了方便预览效果，用户可以使用移动设备模拟器，常见的移动设备模拟器是 Opera Mobile Emulator。

Opera Mobile Emulator 是一款针对电脑桌面开发的模拟移动设备的浏览器，几乎完全重现 opera mobile 手机浏览器的使用效果，可自行设置需要模拟的不同型号的手机和平板电脑配置，然后电脑上模拟各类手机等移动设备访问网站。

Opera Mobile Emulator 的下载网址：http://www.opera.com/zh-cn/developer/mobile-emulator/，根据不同的系统选择不同的版本，这里选择 windows 系统下的版本，如图 17-1 所示。

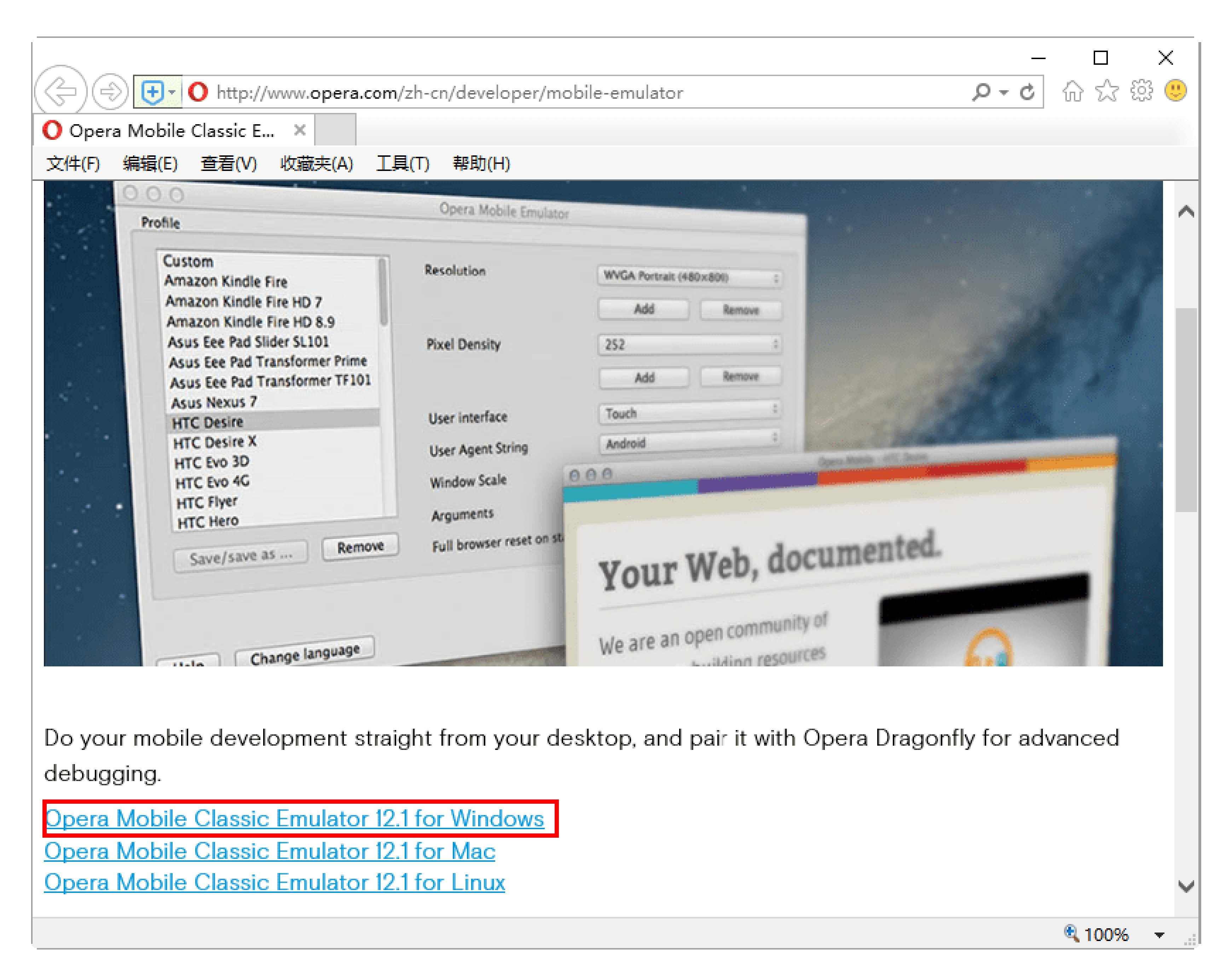


图 17-1 Opera Mobile Emulator 的下载页面

下载并安装之后启动 Opera Mobile Emulator，打开如图 17-2 所示的窗口，在【资料】列表框中选择移动设备的类型，这里选择【LG Optimus 3D】选项，单击【启动】按钮。

打开欢迎界面，用户可以单击不同的链接，查看该软件的功能，如图 17-3 所示。

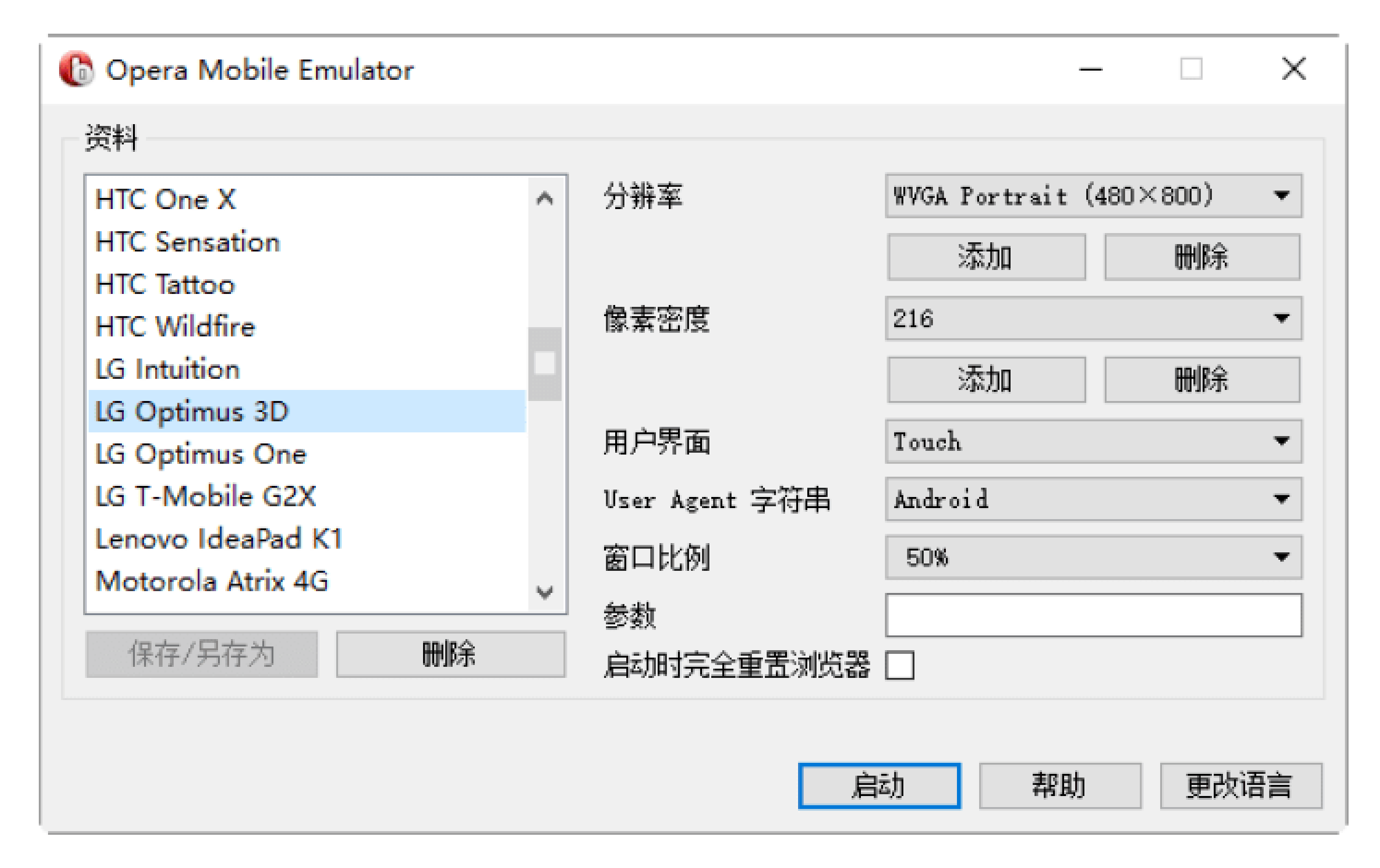


图 17-2　参数设置

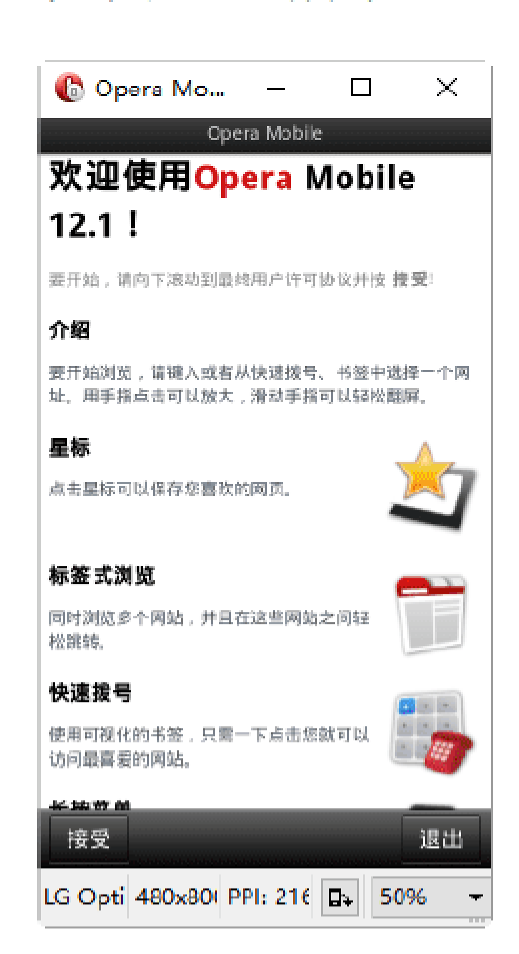


图 17-3　欢迎界面

单击【接受】按钮，打开手机模拟器界面，在【输入网站】文本框中输入需要查看网页效果的地址即可，如图 17-4 所示。

例如这里直接单击【当当网】图标，即可查看当当网在该移动设备模拟器中的效果，如图 17-5 所示。



图 17-4　手机模拟器界面

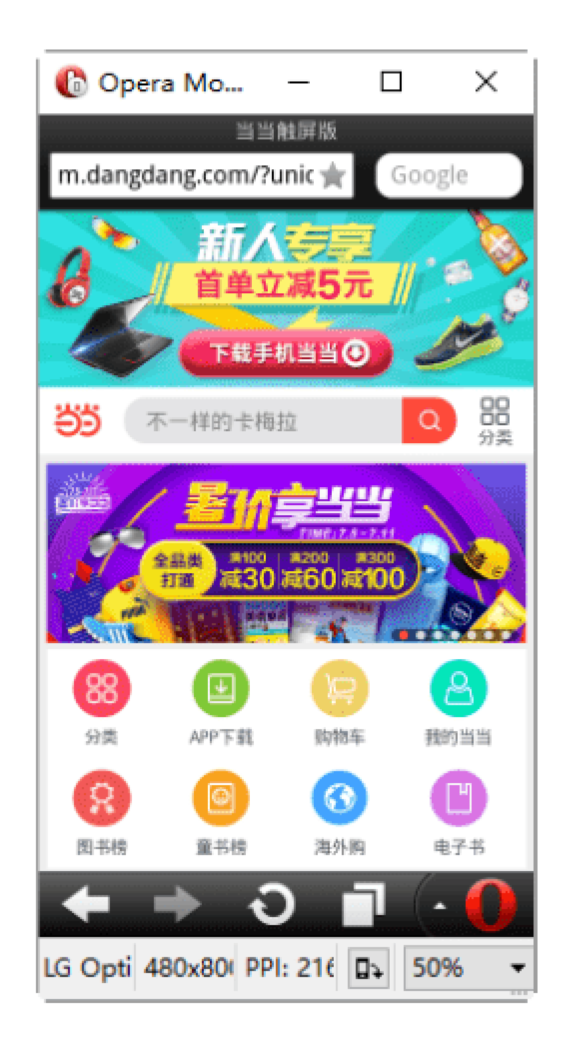


图 17-5　查看预览效果

Opera Mobile Emulator 不仅可以查看移动网页的效果，还可以任意调整窗口的大小，从而可以查看不用屏幕尺寸的效果，这点也是 Opera Mobile Emulator 与其他移动设备模拟器相比最大的优势。

17.2.2 jQuery Mobile 的安装

想要开发 jQuery Mobile 网页，必须要引用 JavaScript 函数库（.js）、CSS 样式表和配套的 jQuery 函数库文件。常见的引用方法有以下两种：

1. 直接引用 jQuery Mobile 库文件

从 jQuery Mobile 的官网下载该库文件（网址是 http://jquerymobile.com/download/），如图 17-6 所示。

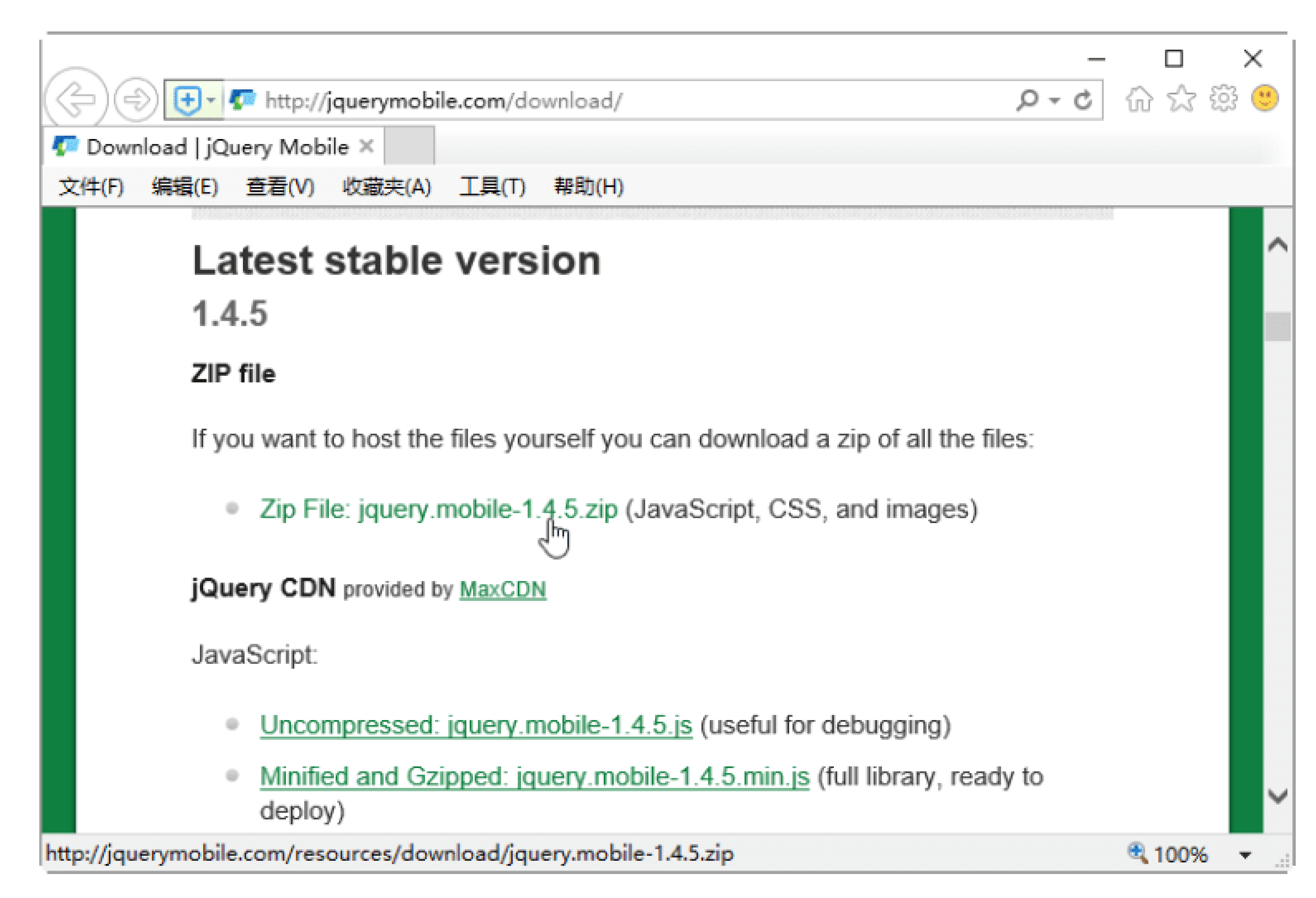


图 17-6 下载 jQuery Mobile 库文件

下载完成即可解压，然后直接引用文件即可，代码如下：

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="jquery.mobile-1.4.5.css">
<script src="jquery.js"></script>
<script src="jquery.mobile-1.4.5.js"></script>
</head>
```

注意，将下载的文件解压到和网页位于同一目录下，否则会无法引用而报错。

细心的读者会发现，在<script>标签中没有插入 type="text/javascript"，这是什么原因呢？因为所有的浏览器中 HTML5 的默认脚本语言就是 JavaScript，所以在 HTML5 中已经不再需要该属性。

2. 从 CDN 中加载 jQuery Mobile

CDN 的全称是 Content Delivery Network，即内容分发网络。其基本思路是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳定。

使用 CDN 中加载 jQuery Mobile，用户不需要在电脑上安装任何东西。用户仅仅需要在网页中加载层叠样式(.css)和 JavaScript 库 (.js) 就能够使用 jQuery Mobile。

用户可以从 jQuery Mobile 官网中查找引用路径，网址是：http://jquerymobile.com/download/，进入该网站后，找到 jQuery Mobile 的引用链接，然后将其复制后添加到 HTML 文件<head>标记中即可，如图 17-7 所示。

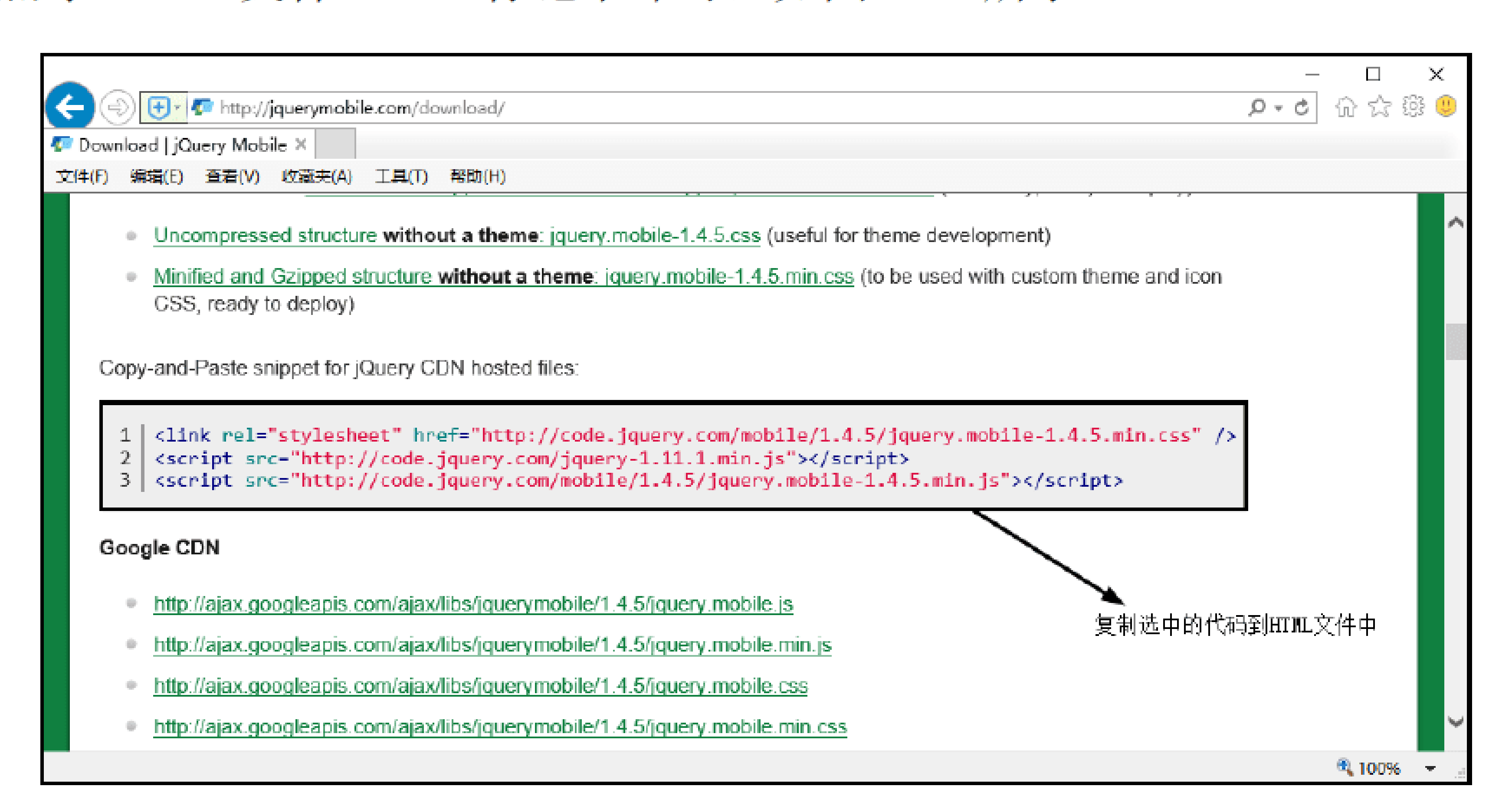


图 17-7 下载 jQuery Mobile 库文件

将代码复制到<head>标记块内。

```
<head>
<!-- meta使用viewport以确保页面可自由缩放 -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- 引入 jQuery Mobile 样式 -->
 <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<!-- 引入 jQuery 库 -->
 <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<!-- 引入 jQuery Mobile 库 -->
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
 </head>
```

注意，因为 jQuery Mobile 函数库仍然在开发中，所以引用的链接中的版本号可能会与本书不同，请使用官方提供的最新版本，只要按照上述方法将代码复制下来引用即可。

17.2.3 jQuery Mobile 网页的架构

jQuery Mobile 网页是由 header、content 与 footer 三个区域组成的架构，利用<div>标记加上 HTML5 自定义属性 data-*来定义移动设备网页组件样式，最基本的属性 data-role 可以用来

定义移动设备的页面架构，语法格式如下：

```
<div data-role="page">
<!—开始一个page-->
  <div data-role="header">
    <h1>这个是标题</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>这里是内容</p>
  </div>
  <div data-role="footer">
    <h1>底部文本</h1>
  </div>
</div>
```

模拟器中预览效果如图 17-8 所示。

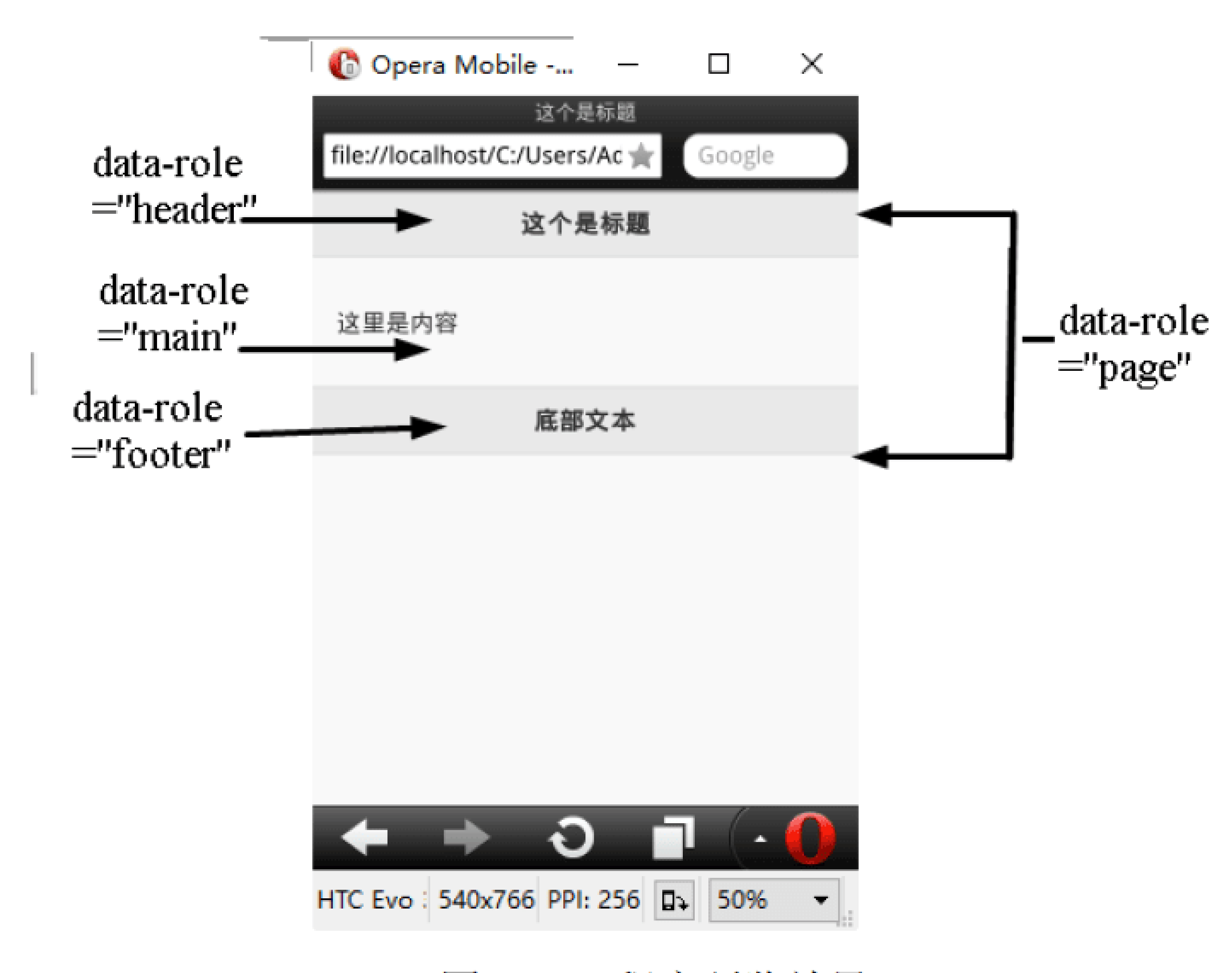


图 17-8　程序预览效果

从结果可以看出，jQuery Mobile 网页以页（page）为单位，一个 HTML 页面可以放一个页面，也可以放多个页面，只是浏览器每次只会显示一页，如果有多个页面，则需要在页面中添加超链接，从而实现多个页面的切换。

【案例分析】

（1）data-role="page" 是在浏览器中显示的页面。

（2）data-role="header" 是在页面顶部创建的工具条，通常用于标题或搜索按钮。

（3）data-role="main" 定义了页面的内容，比如文本、图片、表单、按钮等。

（4）"ui-content" 类用于在页面添加内边距和外边距。

（5）data-role="footer" 用于创建页面底部工具条。

17.3 创建多页面的 jQuery Mobile 网页

本实例将使用 jQuery Mobile 制作一个多页面的 jQuery Mobile 网页，并创建多个页面。使用不同的 id 属性来区分不同的页面。

【例 17.1】（实例文件：ch17\17.1.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>几回花下坐吹箫，银汉红墙入望遥。</p>
    <a href="#second">下一页</a>
  </div>
  <div data-role="footer">
    <h1>清代诗人</h1>
  </div>
</div>
<div data-role="page" id="second">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>似此星辰非昨夜，为谁风露立中宵。</p>
    <a href="#first">上一页</a>
  </div>
  <div data-role="footer">
    <h1>清代诗人</h1>
  </div>
</div>
</body>
</html>
```

模拟器中预览效果如图 17-9 所示。单击【下一页】超链接，即可进入第二页，如图 17-10 所示。单击【上一页】超链接，即可返回到第一页中。




图 17-9　程序预览效果



图 17-10　第二页预览效果

17.4　将页面作为对话框使用

对话框是用于显示页面信息显示或者表单信息的输入。jQuery Mobile 通过在链接中添加如下属性，即可将页面作为对话框使用。

```
    data-dialog="true"
```

【例 17.2】（实例文件：ch17\17.2.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
    </head>
    <body>
    <div data-role="page" id="first">
      <div data-role="header">
        <h1>古诗鉴赏</h1>
      </div>
```

```
    <div data-role="main" class="ui-content">
      <p>浩荡离愁白日斜，吟鞭东指即天涯。落红不是无情物，化作春泥更护花。</p>
      <a href="#second">查看详情</a>
    </div>
    <div data-role="footer">
      <h1>清代诗词</h1>
    </div>
  </div>
  <div data-role="page" data-dialog="true" id="second">
    <div data-role="header">
      <h1>诗词鉴赏</h1>
    </div>
    <div data-role="main" class="ui-content">
      <p>这首诗是《己亥杂诗》的第五首，写诗人离京的感受。虽然载着“浩荡离愁”，却表示仍然要为国为民尽自己最后一份心力。</p>
      <a href="#first">上一页</a>
    </div>
    <div data-role="footer">
      <h1>清代诗词</h1>
    </div>
  </div>
  </body>
  </html>
```

模拟器中预览效果如图 17-11 所示。单击【查看详情】超链接，即可打开一个对话框，如图 17-12 所示。



图 17-11　程序预览效果



图 17-12　对话框预览效果

从结果可以看出，对话框与普通页面不同，它显示在当期页面上，但又不会填充完整的页面，顶部图标 用于关闭对话框，单击【上一页】链接也可以关闭对话框。

17.5 绚丽多彩的页面切换效果

jQuery Mobile 提供了各种页面切换到下一个页面的效果，主要通过设置 data-transition 属性来完成各种页面切换效果。语法规则如下：

```
<a href="#link" data-transition="切换效果">切换下一页</a>
```

其中切换效果有很多，如表 17-1 所示。

表 17-1 页面切换效果

页面效果参数	含义
fade	默认的切换效果。淡入到下一页
none	无过渡效果
flip	从后向前翻转到下一页
flow	抛出当前页，进入下一页
pop	像弹出窗口那样转到下一页
slide	从右向左滑动到下一页
slidefade	从右向左滑动并淡入到下一页
slideup	从下到上滑动到下一页
slidedown	从上到下滑动到下一页
turn	转向下一页

注意，在 jQuery Mobile 的所有链接上，默认使用淡入淡出的效果。

例如，设置页面从右向左滑动到下一页，代码如下：

```
<a href="#second" data-transition="slide">切换下一页</a>
```

上面的所有效果支持后退行为。例如，用户想让页面从左向右滑动，可以添加 data-direction 属性为"reverse" 值即可。代码如下：

```
<a href="#second" data-transition="slide" data-direction="reverse">切换下一页</a>
```

【例 17.3】（实例文件：ch17\17.3.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
```

```
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
     </head>
    <body>
    <div data-role="page" id="first">
      <div data-role="header">
        <h1>古诗欣赏</h1>
      </div>
      <div data-role="main" class="ui-content">
        <p>老农家贫在山住，耕种山田三四亩。</p>
    <!—实现从右到左切换到下一页 -->
        <a href="#second" data-transition="slide" >下一页</a>
      </div>
      <div data-role="footer">
        <h1>野老歌</h1>
      </div>
    </div>
    <div data-role="page" id="second">
      <div data-role="header">
        <h1>古诗欣赏</h1>
      </div>
      <div data-role="main" class="ui-content">
        <p>岁暮锄犁傍空室，呼儿登山收橡实。</p>
    <!—实现从左到右切换到下一页 -->
        <a href="#first" data-transition="slide" data-direction="reverse">上
一页</a>
      </div>
      <div data-role="footer">
        <h1>野老歌</h1>
      </div>
    </div>
    </body>
    </html>
```

模拟器中预览效果如图 17-13 所示。单击【下一页】超链接即可从右到左滑动进入第二页，如图 17-14 所示。单击【上一页】超链接即可从左到右滑动返回到第一页中。



图 17-13 程序预览效果



图 17-14 第二页预览效果

17.6 专家解惑

1. 如何在模拟器中查看做好的网页效果？

HTML 文件制作完成后，要想在模拟器中测试，可以在地址栏中输入文件的路径，例如输入如下：

```
file://localhost/ch16/16.2.html
```

为了防止输入错误，可以直接将文件拖曳到地址栏中，模拟器会自动帮助用户添加完整路径。

2. jQuery Moblie 都是支持哪些移动设备？

目前市面上移动设备非常多，如果想查询 jQuery Moblie 都是支持哪些移动设备，可以参照 jQuery Moblie 网站的各厂商支持表，还可以参考维基百科网站对 jQuery Moblie 说明中提供的 Mobile browser support 一览表。

3. 我的浏览器为什么不支持页面切换效果？

为了实现页面切换效果，浏览器必须支持功能。目前，支持 CSS3 3D 切换功能的浏览器最小版本包括 IE 10.0 浏览器、Chrome 12.0 浏览器、Firefox 16.0 浏览器、Safari 4.0 浏览器和 Opera 15.0 浏览器等。

第18章　jQuery Mobile UI组件

jQuery Mobile 针对用户界面提供了各种可视化的标签，包括按钮、复选框、选择菜单、列表、弹窗、工具栏、面板、导航和布局等，这些可视化标签与 HTML5 标记一起使用，即可轻松地开发出绚丽多彩的移动网页。本章节将重点学习这些标签的使用方法和技巧。

18.1　套用 UI 组件

jQuery Mobile 提供很多可视化的 UI 组件，只要套用之后，就可以生成绚丽并且适合移动设备使用的组件。jQuery Mobile 中各种可视化的 UI 组件与 HTML5 标记大同小异。下面介绍常用的组件的用法，其中按钮、列表等功能变化比较的大的后面会做详细介绍。

18.1.1　表单组件

jQuery Mobile 使用 CSS 自动为 HTML 表单添加样式，让它们看起来更具吸引力，触摸起来更具友好性。

在 jQuery Mobile 中，经常使用的表单控件如下：

1. 文本框输入框

文本输入框的语法规则如下：

```
<input type="text" name="fname" id="fname" value=" ">
```

其中 value 属性是文本框中显示的内容，也可以使用 placeholder 来指定一个简短的描述，用来描述输入内容的含义。

【例 18.1】（实例文件：ch18\18.1.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
```

```
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
     </head>
    <body>
    <div data-role="first">
      <div data-role="header">
      <h1>输入会员信息</h1>
      </div>
      <div data-role="main" class="ui-content">
        <form>
          <div class="ui-field-contain">
            <label for="fullname">姓名: </label>
            <input type="text" name="fullname" id="fullname">
            <label for="bday">出生年月: </label>
            <input type="date" name="bday" id="bday">
            <label for="email">E-mail:</label>
            <input type="email" name="email" id="email" placeholder="输入您的电
子邮箱">
          </div>
          <input type="submit" data-inline="true" value="注册">
        </form>
      </div>
    </div>
    </body>
    </html>
```

模拟器中预览效果如图 18-1 所示。单击【出生年月】下拉按钮时，会自动打开日期选择器，用户直接选择相应的日期即可，如图 18-2 所示。



图 18-1　程序预览效果



图 18-2　日期选择器

2. 文本域

使用<textarea>可以实现多行文本输入效果。

【例 18.2】（实例文件：ch18\18.2.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
     </head>
    <body>
    <div data-role="first">
      <div data-role="header">
      <h1>文本框</h1>
      </div>
      <div data-role="main" class="ui-content">
        <form>
           <div class="ui-field-contain">
            <label for="info">输入最喜欢的一首古诗:</label>
            <textarea name="addinfo" id="info"></textarea>
          </div>
          <input type="submit" data-inline="true" value="提交">
        </form>
         </div>
    </div>
    </body>
    </html>
```

模拟器中预览效果如图 18-3 所示。输入多行内容时，文本框会根据输入的内容，自动调整文本框的高度，如图 18-4 所示。



图 18-3　程序预览效果

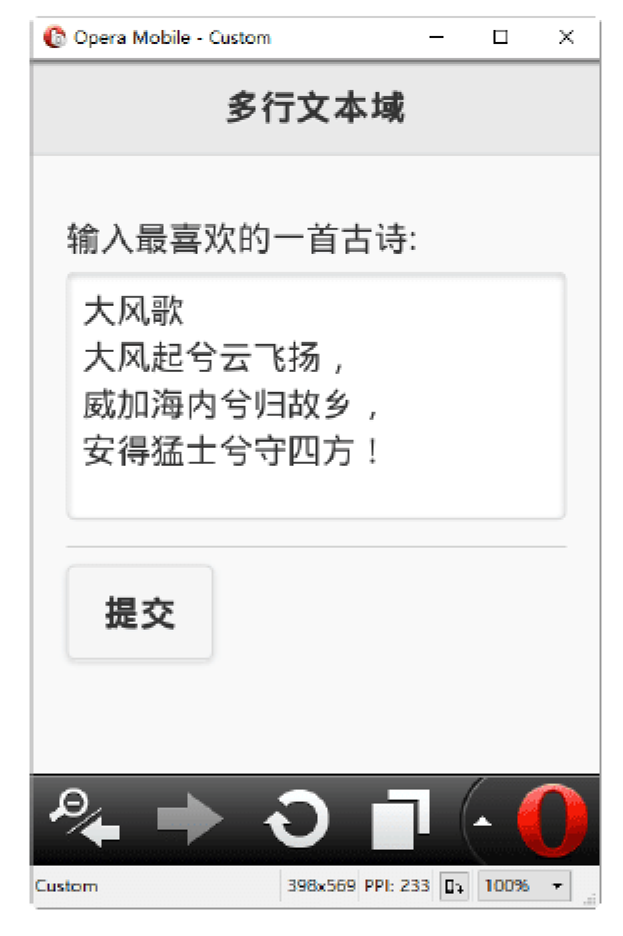


图 18-4　选择日期

3. 搜索输入框

HTML5 中新增的 type="search"类型为搜索输入框，它是为输入搜索定义文本字段。

搜索输入框的语法规则如下：

```
<input type="search" name="search" id="search" placeholder="搜索内容">
```

搜索输入框的效果如图 18-5 所示。

图 18-5　搜索输入框

4. 范围滑动条

使用<input type="range">控件，即可创建范围滑动条，语法格式如下：

```
<input type="range" name="points" id="points" value="50" min="0" max="100"
data-show-value="true">
```

其中，max 属性规定允许的最大值；min 属性规定允许的最小值；step 属性规定合法的数字间隔；value 属性规定默认值；data-show-value 属性规定是否在按钮上显示进度的值，如果设置为 true，则表示显示进度的值，如果设置为 false，则表示不显示进度的值。

【例 18.3】（实例文件：ch18\18.3.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
 </head>
<body>
<div data-role="first">
  <div data-role="header">
    <h1>工作进度申报</h1>
  </div>
  <div data-role="main" class="ui-content">
    <form>
      <label for="points">工作完成的进度:</label>
      <input type="range" name="points" id="points" value="50" min="0"
max="100" data-show-value="true">
```

```
        <input type="submit" data-inline="true" value="提交">
      </form>
    </div>
  </div>
  </body>
  </html>
```

模拟器中预览效果如图 18-6 所示。用户可以拖动滑块，选择需要的值。也可以通过加减按钮，精确选择进度的值。



图 18-6　程序预览效果

使用 data-popup-enabled 属性可以设置小弹窗效果，代码如下：

```
  <input type="range" name="points" id="points" value="50" min="0" max="100"
data-popup-enabled="true">
```

添加后的效果如图 18-7 所示。



图 18-7　进度值显示效果

使用 data-highlight 属性可以亮度显示滑动条的值。代码如下：

```
  <input type="range" name="points" id="points" value="50" min="0" max="100"
data-highlight="true">
```

添加后的效果如图 18-8 所示。

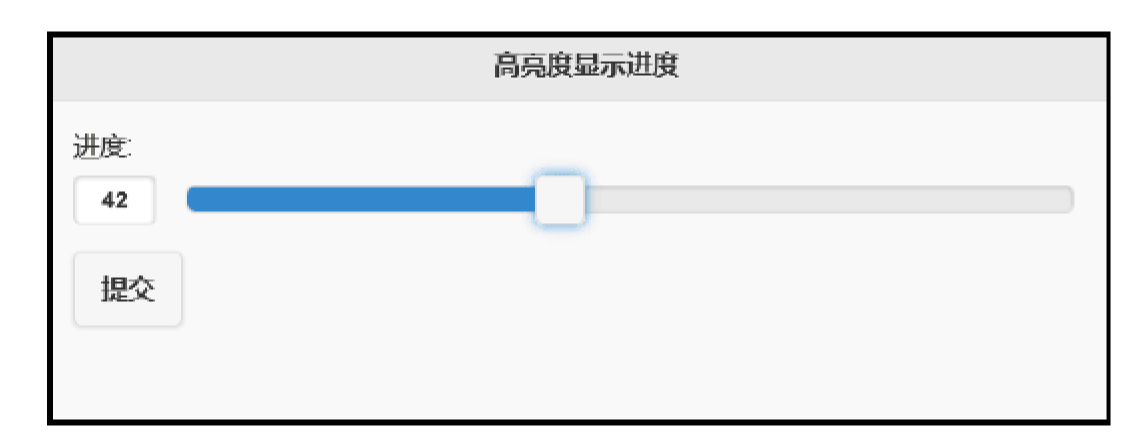


图 18-8　高亮度显示进度值效果

5. 表单按钮

表单按钮分为三种，普通按钮，提交按钮和取消按钮。只需要在 type 属性中设置表单的类型即可，代码如下：

```
<input type="submit" value="提交按钮">
<input type="reset" value="取消按钮">
<input type="button" value="普通按钮">
```

模拟器中预览效果如图 18-9 所示。



图 18-9　表单按钮预览效果

当用户在有限数量的选择中仅选取一个选项时，经常用到表单中的单选按钮。通过 type="radio"来创建一系列的单选按钮，代码如下：

```
<fieldset data-role="controlgroup">
<legend>请选择您的年级：</legend>
  <label for="one">一年级</label>
  <input type="radio" name="grade" id="one" value="one">
  <label for="two">二年级</label>
  <input type="radio" name="grade" id="two" value="two">
  <label for="three">三年级</label>
  <input type="radio" name="grade" id=" three" value=" three">
</fieldset>
```

模拟器中预览效果如图 18-10 所示。



图 18-10　单选按钮

提示：<fieldset>标记用来创建按钮组，组内各个组件保持自己的功能。在<fieldset>标记内添加 data-role="controlgroup"，这样这些单选按钮样式统一，看起来像一个组合。其中<legend>标签来定义按钮组的标题

6. 复选框

当用户在有限数量的选择中选取一个或多个选项时，需要使用复选框，代码如下：

```
<fieldset data-role="controlgroup">
   <legend>请选择您喜爱的季节：</legend>
   <label for="spring">春天</label>
   <input type="checkbox" name="season" id="spring" value="spring">
   <label for="summer">夏天</label>
   <input type="checkbox" name="season" id="summer" value="summer">
   <label for="fall">秋天</label>
   <input type="checkbox" name="season" id="fall" value="fall">
   <label for="winter">冬天</label>
   <input type="checkbox" name="season" id="winter" value="winter">
 </fieldset>
```

模拟器中预览效果如图 18-11 所示。



图 18-11　复选框

6. 选择菜单

使用<select>标签可以创建带有若干选项的下拉列表。<select>标签内的<option>属性定义

了列表中的可用选项，代码如下：

```
<fieldset data-role="fieldcontain">
      <label for="day">选择值日时间：</label>
      <select name="day" id="day">
       <option value="mon">星期一</option>
       <option value="tue">星期二</option>
       <option value="wed">星期三</option>
       <option value="thu">星期四</option>
       <option value="fri">星期五</option>
       <option value="sat">星期六</option>
       <option value="sun">星期日</option>
      </select>
</fieldset>
```

模拟器中预览效果如图 18-12 所示。



图 18-12　选择菜单

如果菜单中的选项还需要再次分组，就可以在<select>内使用<optgroup>标签。添加后的代码如下：

```
<fieldset data-role="fieldcontain">
      <label for="day">选择值日时间：</label>
      <select name="day" id="day">
      <optgroup label="工作日">
       <option value="mon">星期一</option>
       <option value="tue">星期二</option>
       <option value="wed">星期三</option>
       <option value="thu">星期四</option>
       <option value="fri">星期五</option>
      </optgroup>
      <optgroup label="休息日">
       <option value="sat">星期六</option>
       <option value="sun">星期日</option>
      </optgroup>
```

```
        </select>
</fieldset>
```

模拟器中预览效果如图 18-13 所示。



图 18-13　菜单选项分组后的效果

如果想选择菜单中的多个选项，就需要设置<select>标签的 multiple 属性，设置代码如下：

```
<select name="day" id="day" multiple data-native-menu="false">
```

例如把上面的代码修改如下：

```
<fieldset data-role="fieldcontain">
        <label for="day">选择值日时间：</label>
        <select name="day" id="day" multiple data-native-menu="false">
        <optgroup label="工作日">
         <option value="mon">星期一</option>
         <option value="tue">星期二</option>
         <option value="wed">星期三</option>
         <option value="thu">星期四</option>
         <option value="fri">星期五</option>
        </optgroup>
        <optgroup label="休息日">
         <option value="sat">星期六</option>
         <option value="sun">星期日</option>
        </optgroup>
        </select>
</fieldset>
```

模拟器中预览，选择菜单时的效果如图 18-14 所示。选择完成后，即可看到多个菜单现象被选择，如图 18-15 所示。

选择多个值日时间：
工作日
星期一
星期二
星期三
星期四
星期五
休息日
星期六
星期日

图 18-14 厦门站多个菜单选项

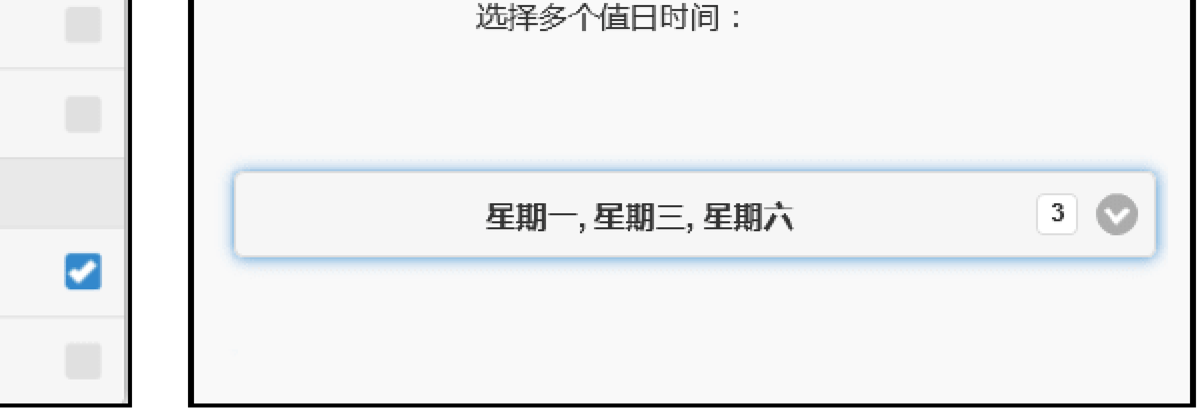


图 18-15 多个菜单选项被选择后的效果

8. 翻转波动开关

设置<input type="checkbox">标签的 data-role 为"flipswitch"时，可以创建翻转波动开关。代码如下：

```
    <form>
       <label for="switch">切换开关：</label>
        <input type="checkbox" data-role="flipswitch" name="switch"
id="switch">
    </form>
```

模拟器中预览效果如图 18-16 所示。

同时，用户还可以使用"checked"属性来设置默认的选项。代码如下：

```
    <input type="checkbox" data-role="flipswitch" name="switch" id="switch"
checked>
```

修改后预览效果如图 18-17 所示。



图 18-16 开关默认效果



图 18-17 修改默认选项后的效果

默认情况下，开关切换的文本为"On"和"Off"。可以使用 data-on-text 和 data-off-text 属性来修改，代码如下：

```
    <input type="checkbox" data-role="flipswitch" name="switch" id="switch"
data-on-text="打开" data-off-text="关闭">
```

修改后预览效果如图 18-18 所示。



图 18-18　修改切换开关文本后的效果

18.1.2　按钮和按钮组

前面简单介绍过表单按钮，由于按钮和按钮组功能变化比较大，本节将详细讲述它们的使用方法和技巧。

在 jQuery Mobile 中，创建按钮的方法包括以下 3 种：

（1）使用<button>标签创建普通按钮。代码如下：

```
<button>按钮</button>
```

（2）使用<input>标签创建表单按钮。代码如下：

```
<input type="button" value="按钮">
```

（3）使用 data-role="button"属性创建链接按钮。代码如下：

```
<a href="#" data-role="button">按钮</a>
```

在 jQuery Mobile 中，按钮的样式会被自动添加上，为了让按钮在移动设备上更具吸引力和可用性。推荐在页面间进行链接时，使用第三种方法;在表单提交时，用第一种或第二种方法。

默认情况下，按钮占满整个屏幕宽度。如果想要一个仅是与内容一样宽的按钮，或者需要并排显示两个或多个按钮，可以通过设置 data-inline="true"来完成。代码如下：

```
<a href="#pagetwo" data-role="button" data-inline="true">下一页</a>
```

下面通过一个案例来区别默认按钮和设置后按钮的区别，代码如下：

【例 18.4】（实例文件：ch18\18.4.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
```

```
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
     </head>
    <body>
    <div data-role="page" id="first">
      <div data-role="header">
        <h1>按钮的区别</h1>
      </div>
      <div data-role="content" class="content">
        <p>普通 / 默认按钮:</p>
        <a href="#second" data-role="button">下一页</a>
        <p>设置后的按钮:</p>
        <a href="#second" data-inline="true">下一页</a>
        <a href="#first" data-inline="true">上一页</a>
      </div>
      <div data-role="footer">
        <h1>2种按钮</h1>
      </div>
    </div>
    </body>
    </html>
```

模拟器中预览效果如图 18-19 所示。

图 18-19　不同的按钮的效果

jQuery Mobile 提供了一个简单的方法来将按钮组合在一起。使用 data-role="controlgroup" 属性即可通过按钮组来组合按钮。同时使用 data-type="horizontal|vertical"属性来设置按钮的排列方式是水平还是垂直。

【例 18.5】（实例文件：ch18\18.5.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
     </head>
    <body>
    <div data-role="page" id="first">
      <div data-role="header">
        <h1>组按钮的排列</h1>
      </div>
      <div data-role="content" class="content">
    <div data-role="controlgroup" data-type="horizontal">
        <p>水平排列的按钮：</p>
        <a href="#" data-role="button">按钮a</a>
        <a href="#" data-role="button">按钮b</a>
        <a href="#" data-role="button">按钮c</a>
    </div><br>
        <div data-role="controlgroup" data-type="vertical"
        <p>垂直排列的按钮:</p>
         <a href="#" data-role="button">按钮a</a>
        <a href="#" data-role="button">按钮b </a>
        <a href="#" data-role="button">按钮c</a>
    </div>
      </div>
      <div data-role="footer">
        <h1>2种排列方式</h1>
      </div>
    </div>
    </body>
    </html>
```

模拟器中预览效果如图 18-20 所示。

图 18-20　不同排列方式的按钮组

18.1.3 按钮图标

jQuery Mobile 提供了一套丰富多彩的按钮图标，用户只需要使用 data-icon 属性即可添加按钮图标，常用的图标样式如表 18.1 所示。

表 18.1 常用的按钮图标样式

图标参数	外观样式	说明
data-icon="arrow-l"	左箭头	左箭头
data-icon="arrow-r"	右箭头	右箭头
data-icon="arrow-u"	上箭头	上箭头
data-icon="arrow-d"	下箭头	下箭头
data-icon="info"	信息	信息
data-icon="plus"	加号	加号
data-icon="minus"	减号	减号
data-icon="check"	复选	复选
data-icon="refresh"	重新整理	重新整理
data-icon="delete"	删除	删除
data-icon="forward"	前进	前进
data-icon="back"	后退	后退
data-icon="star"	星形	星号
data-icon="audio"	扬声器	扬声器
data-icon="lock"	挂锁	挂锁
data-icon="search"	搜索	搜索
data-icon="alert"	警告	警告
data-icon="grid"	网格	网格
data-icon="home"	首页	主页

例如以下代码：

```
<a href="#" data-role="button" data-icon="lock">挂锁</a>
<a href="#" data-role="button" data-icon="check">复选</a>
<a href="#" data-role="button" data-icon="refresh">重新整理</a>
<a href="#" data-role="button" data-icon="delete">删除</a>
```

模拟器中预览效果如图 18-21 所示。



图 18-21　按钮图标效果

细心的读者会发现，按钮上的图标默认情况下会出现在按钮的左边。如果需要设置图标的位置，可以设置 data-iconpos 属性来指定位置，包括 top（顶部）、right（右侧）bottom（底部）。例如以下代码：

```
    <a href="#" data-role="button" data-icon="refresh">重新整理</a>
    <a href="#" data-role="button" data-icon="refresh" data-iconpos="top">重
新整理</a>
    <a href="#" data-role="button" data-icon="refresh" data-iconpos="right">
重新整理</a>
    <a href="#" data-role="button" data-icon="refresh" data-iconpos="bottom">
重新整理</a>
```

模拟器中预览效果如图 18-22 所示。



图 18-22　设置图标的位置



如果不想让按钮上出现文字，就可以将 data-iconpos 属性设置为 notext，这样只会显示按钮而没有文字。

18.1.4　弹窗

弹窗是一个非常流行的对话框，弹窗可以覆盖在页面上展示。弹窗可用于显示一段文本，图片，地图或其他内容。创建一个弹窗，需要使用<a>和<div>标签。在<a>标签上添加 data-rel="popup"属性，<div>标签添加 data-role="popup"属性。然后为<div>设置 id，设置<a>的

href 值为<div>指定的 id，其中<div>中的内容为弹窗显示的内容。代码如下：

```
<a href="#firstpp" data-rel="popup">显示弹窗</a>
<div data-role="popup" id="firstpp">
  <p>这是弹出窗口显示的内容</p>
</div>
```

模拟器中预览效果如图 18-23 所示。单击【显示弹窗】即可显示弹出窗口的内容。



图 18-23　弹窗效果

注意: <div>弹窗与点击的<a>链接必须在同一个页面上。

默认情况下，点击弹窗之外的区域或按下【Esc】键即可关闭弹窗。用户也可以在弹窗上添加关闭按钮，只需要设置属性 data-rel="back"即可，结果如果 18-24 所示。



图 18-24　带关闭按钮的弹窗效果

用户还可以在弹窗中显示图片。代码如下：

```
<div id="pageone" data-role="content" class="content" >
    <p>单击下面的小图片</p>
    <a href="#firstpp" data-rel="popup" >
    <img src="123.jpeg" style="width:200px;"></a>
    <div data-role="popup" id="firstpp">
    <p>这是我的图片！</p>
    </a><img src="123.jpeg" style="width:500px;height:500px;" >
    </div>
  </div>
```

模拟器中预览效果如图 18-25 所示。单击图片，即可弹出如图 18-26 所示的图片弹窗。



图 18-25　模拟器中预览效果



图 18-26　图片弹窗效果

18.2 列表

因为移动设备屏幕比较小，所以常常以列表的形式显示数据。本章节将学习列表的使用方法和技巧。

18.2.1 列表视图

jQuery Mobile 中的列表视图是标准的 HTML 列表，包括有序列表<ol>和无序列表<ul>。列表视图是 jQuery Mobile 中功能强大的一个特性。它会使标准的无序或有序列表应用更广泛。

列表的使用方法非常简单，只需要在<ul>或<ol>标签中添加属性 data-role="listview"。每个项目(<li>)中可以添加链接。下面通过一个案例来学习。

【例 18.6】（实例文件：ch18\18.6.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
     </head>
    <body>
    <div data-role="page" id="first">
      <div data-role="header">
        <h1>列表视图</h1>
      </div>
      <div data-role="content" class="content">
      <h2>有序列表：</h2>
        <ol data-role="listview">
          <li><a href="#">香蕉</a></li>
          <li><a href="#">橘子</a></li>
          <li><a href="#">苹果</a></li>
        </ol>
        <h2>无序列表：</h2>
        <ul data-role="listview">
          <li><a href="#">芹菜</a></li>
          <li><a href="#">韭菜</a></li>
          <li><a href="#">菠菜</a></li>
        </ul>
    </div>
      </div>
      <div data-role="footer">
```

```
      <h1>有序列表和无序列表</h1>
    </div>
  </div>
  </body>
  </html>
```

模拟器中预览效果如图 18-27 所示。

图 18-27　有序列表和无序列表



默认情况下，列表项的链接会自动变成一个按钮，此时不再需要使用 data-role="button"属性。

从结果可以看出，列表样式中没有边缘和圆角效果，这里可以通过设置属性 data-inset="true"来完成，代码如下：

```
<ul data-role="listview" data-inset="true">
```

上面案例的部分代码修改如下：

```
<div data-role="content" class="content">
  <h2>标准列表样式：</h2>
    <ol data-role="listview">
      <li><a href="#">香蕉</a></li>
      <li><a href="#">橘子</a></li>
      <li><a href="#">苹果</a></li>
    </ol>
    <h2>添加data-inset="true"属性后的样式：</h2>
    <ul data-role="listview" data-inset="true">
      <li><a href="#">芹菜</a></li>
      <li><a href="#">韭菜</a></li>
      <li><a href="#">菠菜</a></li>
    </ul>
</div>
```

模拟器中预览效果如图 18-28 所示。

标准列表样式：
1. 香蕉
2. 橘子
3. 苹果
添加data-inset="true"属性后的样式：
芹菜
韭菜
菠菜

图 18-28　有边缘和圆角的列表效果

如果列表项比较多，用户可以使用列表分割项对列表进行分组操作，这样使列表看起来更整齐。通过在列表项<li>标签中添加 data-role="list-divider" 属性即可指定列表分割，例如以下代码：

```
<ul data-role="listview">
 <li data-role="list-divider">蔬菜</li>
  <li><a href="#">芹菜</a></li>
  <li><a href="#">韭菜</a></li>
<li data-role="list-divider">水果</li>
  <li><a href="#">苹果</a></li>
  <li><a href="#">橘子</a></li>
<li data-role="list-divider">乳制品</li>
  <li><a href="#">酸奶</a></li>
  <li><a href="#">奶酪</a></li>
 </ul>
```

模拟器中预览效果如图 18-29 所示。

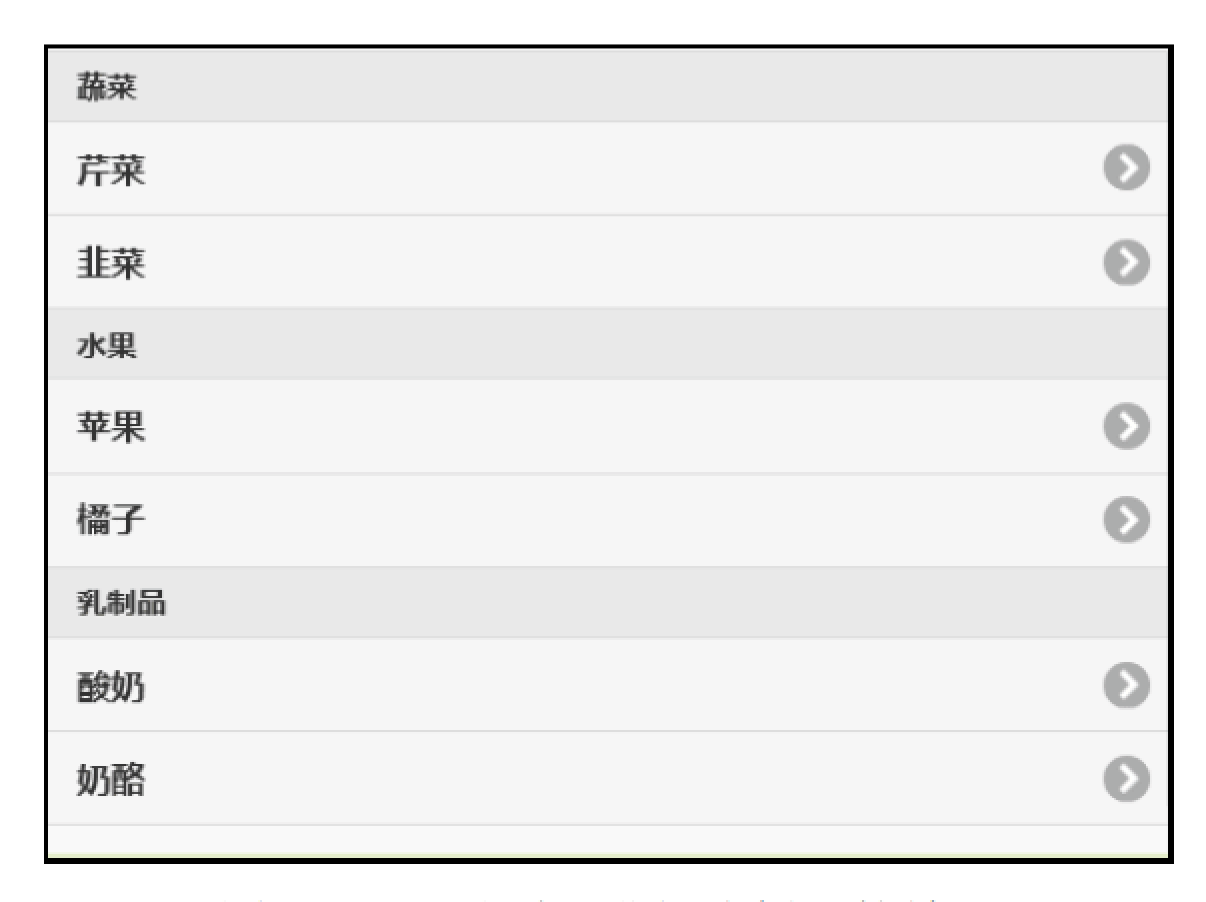


图 18-29　对项目进行分割后的效果

如果项目列表是一个按字母顺序排列的列表，通过添加 data-autodividers="true"属性，可以自动生成项目的分割，代码如下：

```
<ul data-role="listview" data-autodividers="true">
  <li><a href="#">Avocado</a></li>
  <li><a href="#"> Apricot</a></li>
  <li><a href="#">Banana</a></li>
  <li><a href="#">Bramley</a></li>
  <li><a href="#"> Cherry </a></li>
</ul>
```

模拟器中预览效果如图 18-30 所示。从结果可以看出，创建的分隔文本是列表项文本的第一个大写字母。

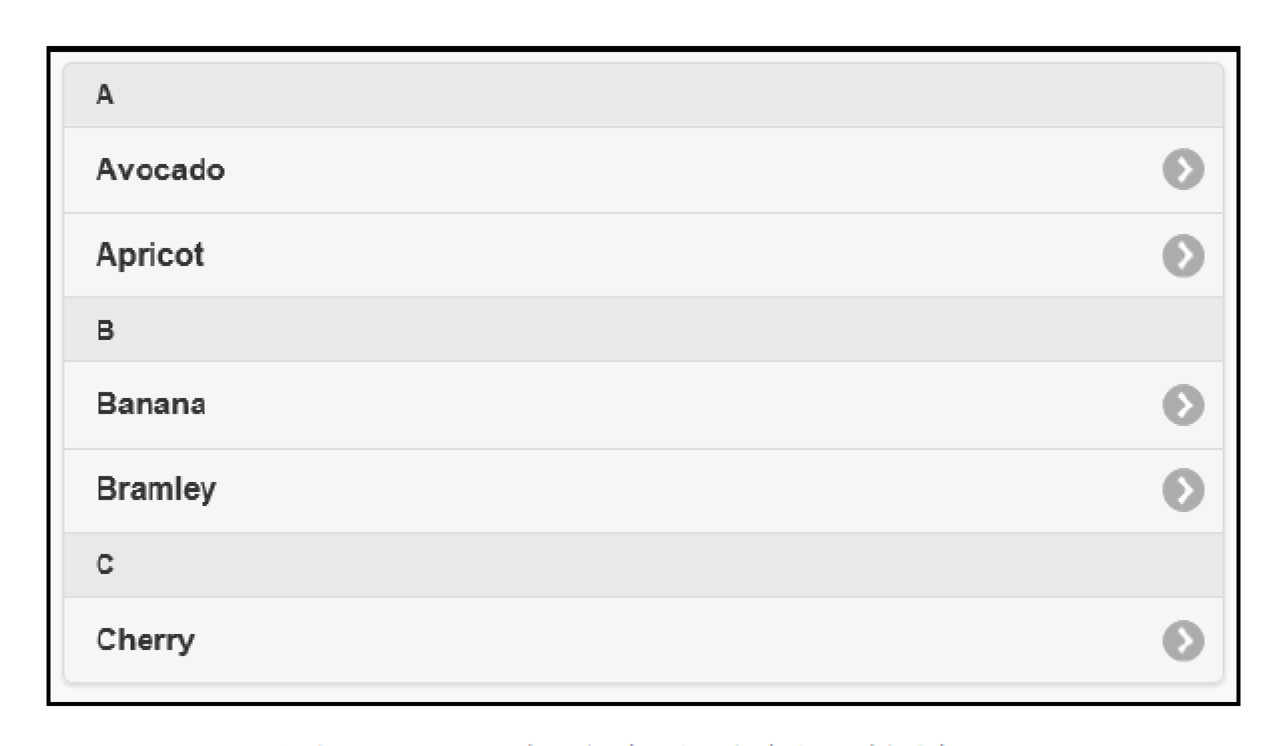


图 18-30　自动生成分割后的效果

18.2.2　列表内容

在列表内容中，既可以添加图片和说明、也可以添加计数泡泡，同时还能拆分按钮和列表的链接。

1. 加入图片和说明

前面做的案例中，列表项目前没有图片或说明。下面来讲述如何添加图片和说明，代码如下：

```
<li>
<a href="#">
<img src="124.jpg">
<h3>香蕉</h3>
<p>香蕉的原产地是东南亚</p>
</a>
</li>
```

模拟器中预览效果如图 18-31 所示。

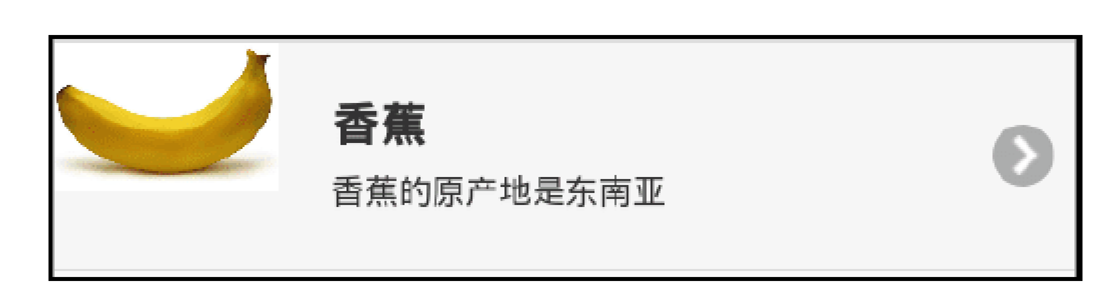


图 18-31　加入图片和说明

2. 计入计数泡泡

计数泡泡主要是在列表中显示数字时使用，只需要在<li>标签加入以下标签即可。

```
<span class="ui-li-count">数字</span>
```

例如下面的例子：

```
<li>
<a href="#">
<img src="124.jpg">
<h3>香蕉</h3>
<p>香蕉的原产地是东南亚</p>
<span class="ui-li-count">111</span>
</a>
</li>
```

模拟器中预览效果如图 18-32 所示。

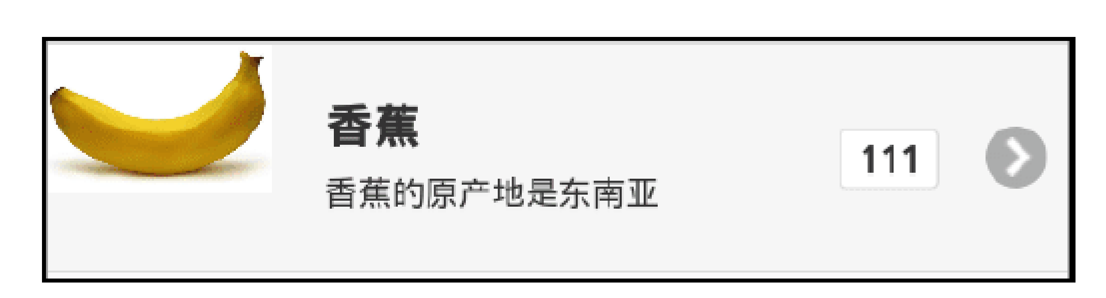


图 18-32　加入计数泡泡

3. 拆分按钮和列表的链接

默认情况下，单击列表项或按钮，都是转向同一个链接。用户也可以拆分按钮和列表项的链接，这样单击按钮和列表项时，会转向不同的链接。设置方法比较简单，值需要在<li>标签中加入两组<a>标签即可。

例如：

```
<li>
<a href="122.html">
<img src="124.jpg">
<h3>香蕉</h3>
<p>香蕉的原产地是东南亚</p>
</a>
<a href="123.html data-icon="star"></a>
</li>
```

模拟器中预览效果如图 18-33 所示。

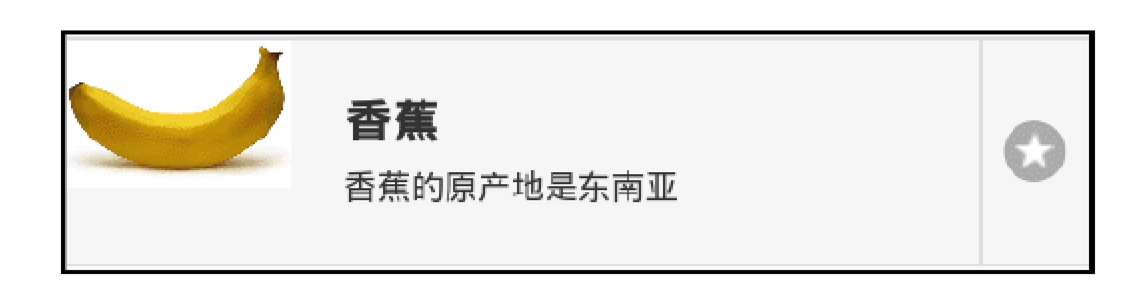


图 18-33　拆分按钮和列表的链接

18.2.3　列表过滤

在 jQuery Mobile 中，用户可以对列表项目进行搜索过滤。添加过滤效果的思路如下：

（1）创建一个表单，并添加类"ui-filterable"，该类的作用是自动调整搜索字段与过滤元素的外边距。代码如下：

```
<form class="ui-filterable">
</form>
```

（2）在<form>标签内创建一个<input>标签，并添加 data-type="search"属性，并指定 id，从而创建基本的搜索字段。代码如下：

```
<form class="ui-filterable">
  <input id="myFilter" data-type="search">
</form>
```

（3）为过滤的列表添加 data-input 属性，该值为<input>标签的 id，代码如下：

```
<ul data-role="listview" data-filter="true" data-input="#myFilter">
```

下面通过一个案例来理解列表是如何过滤的。

【例 18.7】（实例文件：ch18\18.7.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="content" class="content">
```

```
    <h2>进货商联系表</h2>
    <form>
     <input id="myFilter" data-type="search">
    </form>
     <ul data-role="listview" data-filter="true" data-input="#myFilter">
       <li><a href="#">张小名</a></li>
       <li><a href="#">刘名园</a></li>
       <li><a href="#">刘鲲鹏</a></li>
      <li><a href="#">张鹏举</a></li>
      <li><a href="#">张鹏远</a></li>
     </ul>
    </div>
</div>
</body>
</html>
```

模拟器中预览效果如图 18-34 所示。输入需要过滤的关键字，例如这里搜索姓张的进货商，结果如图 18-35 所示。



图 18-34　程序预览效果



图 18-35　列表过滤后的效果

提示：如果需要在搜索框内添加提示信息，就可以通过设置 placeholder 属性来完成。代码如下：

```
<input id="myFilter" data-type="search" placeholder="请输入联系人的姓">
```

18.3　面板和可折叠块

在 jQuery Mobile 中，可以通过面板或可折叠块来隐藏或显示指定的内容。本章节将重点学习面板和可折叠块的使用方法和技巧。

18.3.1　面板

jQuery Mobile 中可以添加面板，面板会在屏幕上从左到右划出。通过为<div>标签添加 data-role="panel"属性来创建面板。具体思路如下：

（1）通过<div>标签来定义面板的内容，并定义 id 属性。例如：

```
<div data-role="panel" id="myPanel">
   <h2>长恨歌</h2>
   <p>天生丽质难自弃，一朝选在君王侧。回眸一笑百媚生，六宫粉黛无颜色。</p>
</div>
```

注意，定义的面板内容必须置于头部、内容和底部组成的页面之前或之后。

（2）要访问面板，需要创建一个指向面板<div>的链接，单击该链接即可打开面板。例如：

```
<a href="#myPanel" class="ui-btn ui-btn-inline">最喜欢的诗句</a>
```

【例 18.8】（实例文件：ch18\18.8.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
 </head>
<body>
<div data-role="first">
  <div data-role="panel" id="myPanel">
    <h2>长恨歌</h2>
    <p>天生丽质难自弃，一朝选在君王侧。回眸一笑百媚生，六宫粉黛无颜色。</p>
  </div>
  <div data-role="header">
  <h1>使用面板</h1>
  </div>
  <div data-role="content" class="content">
     <a href="#myPanel" class="ui-btn ui-btn-inline">最喜欢的诗句</a>
   </div>
</div>
</body>
</html>
```

模拟器中预览效果如图 18-36 所示。单击【最喜欢的诗句】链接，即可打开面板，结果如图 18-37 所示。



图 18-36　程序预览效果

图 18-37　打开面板

面板的展示方式有属性 data-display 来控制，分为以下三种：

(1) data-display="reveal"：面板的展示方式为从左到右划出，这是面板展示方式的默认值。
(2) data-display="overlay"：在内容上显示面板。
(3) data-display="push"：同时"推动"面板和页面。

这三种面板展示方式的代码如下：

```
<div data-role="panel" id="overlayPanel" data-display="overlay">
<div data-role="panel" id="revealPanel" data-display="reveal">
<div data-role="panel" id="pushPanel" data-display="push">
```

默认情况下，面板会显示在屏幕的左侧。如果想让面板出现在屏幕的右侧，可以指定 data-position="right" 属性。

```
<div data-role="panel" id="myPanel" data-position="right">
```

默认情况下，面板是随着页面一起滚动的。如果你需要实现面板内容固定不随页面滚动而滚动，可以在面板添加 the data-position-fixed="true"属性。代码如下：

```
<div data-role="panel" id="myPanel" data-position-fixed="true">
```

18.3.2　可折叠块

通过可折叠块，用户可以隐藏或显示指定的内容，这对于存储部分信息很有用。

创建可折叠块的方法比较简单，只需要在<div>标签添加 data-role="collapsible" 属性即可，添加标题标签为 H1-H6，后面即可添加隐藏的信息。

```
<div data-role="collapsible">
 <h1>折叠块的标题</h1>
 <p>可折叠的具体内容。</p>
 </div>
```

【例 18.9】（实例文件：ch18\18.9.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
     </head>
    <body>
    <div data-role="first">
      <div data-role="header">
      <h1>可折叠块</h1>
      </div>
      <div data-role="content" class="content">
      <div data-role="collapsible">
     <h1>最喜欢的水果</h1>
     <p>香蕉、橘子、苹果</p>
     </div>
      </div>
    </div>
    </body>
    </html>
```

模拟器中预览效果如图 18-38 所示。单击【最喜欢的水果】按钮，即可打开可折叠块，结果如图 18-39 所示。





图 18-38　程序预览效果



图 18-39　打开可折叠块



默认情况下，内容是被折叠起来的。如需在页面加载时展开内容，可以添加 data-collapsed="false"属性，代码如下：

```
<div data-role="collapsible" data-collapsed="false">
 <h1>折叠块的标题</h1>
 <p>这里显示的内容是展开的</p>
 </div>
```

可折叠块是可以嵌套的，例如以下代码：

```
<div data-role="collapsible">
 <h1>全部智能商品</h1>
<div data-role="collapsible">
 <h1>智能家居</h1>
 <p>智能办公、智能厨电和智能网络</p>
 </div>
</div>
```

模拟器中预览效果如图 18-40 所示。

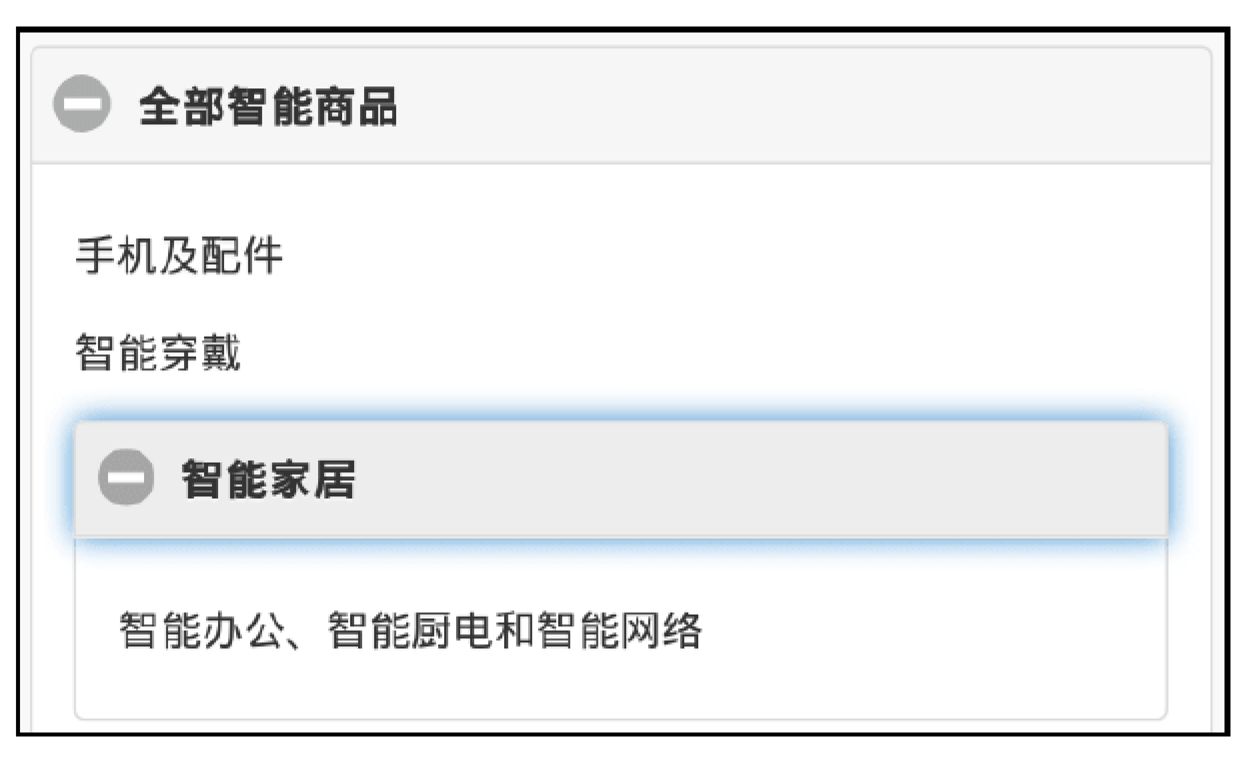


图 18-40　程序预览效果

18.4　导航条

导航条通常位于页面的头部或尾部，主要作用是便于用户快速访问需要的页面。本章节将重点学习导航条的使用方法和技巧。

在 jQuery Mobile 中，使用 data-role="navbar" 属性来定义导航栏。需要特别注意的是，导航栏中的链接将自动变成按钮，不需要使用 data-role="button"属性。

例如以下代码：

```
  <div data-role="header">
    <h1>鸿鹄网购平台</h1>
    <div data-role="navbar">
      <ul>
        <li><a href="#">主页</a></li>
        <li><a href="#">团购</a></li>
        <li><a href="#">搜索商品</a></li>
```

```
      </ul>
    </div>
  </div>
```

模拟器中预览效果如图 18-41 所示。

鸿鹄网购平台

主页 团购 搜索商品

图 18-41 程序预览效果

通过前面章节的学习，用户还可以为导航添加按钮图标，例如以上代码修改如下：

```
  <div data-role="header">
    <h1>鸿鹄网购平台</h1>
    <div data-role="navbar">
      <ul>
        <li><a href="#" data-icon="home">主页</a></li>
        <li><a href="#" data-icon="arrow-d">团购</a></li>
        <li><a href="#" data-icon="search">搜索商品</a></li>
      </ul>
    </div>
  </div>
```

模拟器中预览效果如图 18-42 所示。



图 18-42 程序预览效果

细心的读者会发现，导航按钮的图标默认位置是位于文字的上方，这个普通的按钮图片是不一样的。如果需要修改导航按钮图标的位置，可以通过设置 data-iconpos 属性来指定位置，包括 left（左侧）、right（右侧）bottom（底部）。

例如下面修改导航按钮图标的位置为文本的左侧，代码如下：

```
  <div data-role="header">
    <h1>鸿鹄网购平台</h1>
    <div data-role="navbar" data-iconpos="left">
      <ul>
        <li><a href="#" data-icon="home" >主页</a></li>
        <li><a href="#" data-icon="arrow-d" >团购</a></li>
        <li><a href="#" data-icon="search">搜索商品</a></li>
      </ul>
    </div>
```

```
    </div>
```

模拟器中预览效果如图 18-43 所示。



图 18-43　程序预览效果

注意，和设置普通按钮图标位置不同的是，这里 data-iconpos="left"属性只能添加到<div>标签中，而不能添加到<li>标签中，否则是无效的，读者可以自行检测。

默认情况下，当单击导航按钮时，按钮的样式会发生变换，例如这里单击【搜索商品】导航按钮，发现按钮的底纹颜色变成了蓝色，如图 18-44 所示。



图 18-44　导航按钮的样式变化

如果用户想取消上面的样式变化，可以添加 class="ui-btn-active"属性即可，例如以下代码：

```
<li><a href="#anylink" class="ui-btn-active">首页</a></li>
```

修改完成后，再次单击【首页】导航按钮时，样式不会发生变化。

对于多个页面的情况，往往用户希望显示哪个页面，对应导航按钮处于被选中状态，下面通过一个案例来讲解。

【例 18.10】（实例文件：ch18\18.10.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
     </head>
    <body>
    <div data-role="page" id="first">
```

```
      <div data-role="header">
      <h1>鸿鹄购物平台</h1>
        <div data-role="navbar">
          <ul>
            <li><a href="#" class="ui-btn-active ui-state-persist">主页
</a></li>
            <li><a href="#second">团购</a></li>
            <li><a href="#">搜索商品</a></li>
          </ul>
        </div>
      </div>
      <div data-role="content" class="content">
    <p>这里是首页显示的内容</p>
     </div>
     <div data-role="footer">
        <h1>首页</h1>
      </div>
    </div>

    <div data-role="page" id="second">
      <div data-role="header">
      <h1>鸿鹄购物平台</h1>
        <div data-role="navbar">
          <ul>
            <li><a href="#first">主页</a></li>
            <li><a href="#" class="ui-btn-active ui-state-persist">团购
</a></li>
            <li><a href="#">搜索商品</a></li>
          </ul>
      </div>
      </div>
      <div data-role="content" class="content">
    <p>这里是团购显示的内容</p>
     </div>
     <div data-role="footer">
        <h1>团购页面</h1>
      </div>
    </div>
    </body>
    </html>
```

模拟器中预览效果如图 18-45 所示。此时默认显示首页的内容，【主页】导航按钮处于选中状态。切换到团购页面后，【团购】导航按钮处于选中状态，如图 18-46 所示。




图 18-45 程序预览效果



图 18-46 【团购】导航按钮处于选中状态

18.5 jQuery Mobile 主题

用户在设计移动网站时往往需要配置背景颜色、导航颜色、布局颜色等，这些工作是非常费时的。为此，jQuery Mobile 提供了两种不同的主题样式，每种主题颜色的按钮、导航、内容等颜色都是配置好的，效果也不相同。

这两种主题分别为 a 和 b，通过设置 data-theme 属性来引用主题 a 或 b，代码如下：

```
<div data-role="page" id="first" data-theme="a">
<div data-role="page" id="first" data-theme="b">
```

1. 主题 a

页面为灰色背景、黑色文字；头部与底部均为灰色背景、黑色文字；按钮为灰色背景、黑色文字；激活的按钮和链接为白色文本、蓝色背景；input 输入框中 placeholder 属性值为浅灰色，value 值为黑色。

下面通过一个案例来讲解主题 a 的样式效果。

【例 18.11】（实例文件：ch18\18.11.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
 </head>
<body>
<div data-role="page" id="first" data-theme="a">
  <div data-role="header">
    <h1>古诗鉴赏</h1>
```

```
      </div>

      <div data-role="content " class="content">
        <p>秋风起兮白云飞，草木黄落兮雁南归。兰有秀兮菊有芳，怀佳人兮不能忘。泛楼船兮济汾
河，横中流兮扬素波。</p>
        <a href="#">秋风辞</a>
        <a href="#" class="ui-btn">更多古诗</a>
        <p>唐诗:</p>
        <ul data-role="listview" data-autodividers="true" data-inset="true">
          <li><a href="#">将进酒</a></li>
          <li><a href="#">春望</a></li>
        </ul>
        <label for="fullname">请输入喜欢诗的名字:</label>
          <input type="text" name="fullname" id="fullname" placeholder="诗词名
称..">
        <label for="switch">切换开关:</label>
          <select name="switch" id="switch" data-role="slider">
            <option value="on">On</option>
            <option value="off" selected>Off</option>
          </select>
      </div>

      <div data-role="footer">
        <h1>经典诗歌</h1>
      </div>
    </div>
    </body>
    </html>
```

主题 a 的样式效果如图 18-47 所示。

图 18-47　主题 a 样式效果

2. 主题 b

页面为黑色背景、白色文字；头部与底部均为黑色背景、白色文字；按钮为白色文字、木炭背景；激活的按钮和链接为白色文本、蓝色背景；input 输入框中 placeholder 性值为浅灰色、value 值为白色。

为了对比主题 a 的样式效果，请将上面案例的中代码：

```
<div data-role="page" id="first" data-theme="a">
```

修改如下：

```
<div data-role="page" id="first" data-theme="a">
```

主题 b 的样式效果如图 18-48 所示。

图 18-48　主题 b 样式效果

主题样式 a 和 b 不仅仅可以应用到页面，也可以单独地应用到页面的头部、内容、底部、导航条、按钮、面板、列表、表单等元素上。

例如，将主题样式 b 添加到页面的头部和底部，代码如下：

```
<div data-role="header" data-theme="b"></div>
<div data-role="footer" data-theme="b"></div>
```

将主题样式 b 添加到对话框的头部和底部，代码如下：

```
<div data-role="page" data-dialog="true" id="second">
```

```
  <div data-role="header" data-theme="b"></div>
  <div data-role="footer" data-theme="b"></div>
</div>
```

将主题样式 b 添加到按钮上时，需要使用 class="ui-btn- a|b "来设置按钮颜色为灰色或黑色。例如，将样式 b 的样式应用到按钮上，代码如下：

```
<a href="#" class="ui-btn">灰色按钮(默认)</a>
<a href="#" class="ui-btn ui-btn-b">黑色按钮</a>
```

预览效果如图 18-49 所示。



图 18-49　按钮添加主题后的效果

在弹窗上应用主题样式的代码如下：

```
<div data-role="popup" id="myPopup" data-theme="b">
```

在头部和底部的按钮上也可以添加主题样式，例如以下代码：

```
<div data-role="header">
  <a href="#" class="ui-btn ui-btn-b">主页</a>
  <h1>古诗欣赏</h1>
  <a href="#" class="ui-btn">搜索</a>
</div>

<div data-role="footer">
   <a href="#" class="ui-btn ui-btn-b">上传古诗图文</a>
  <a href="#" class="ui-btn">名句欣赏鉴别</a>
  <a href="#" class="ui-btn ui-btn-b">联系我们</a>
</div>
```

预览效果如图 18-50 所示。



图 18-50　头部和底部的按钮添加主题后的效果

18.6 专家解惑

1. 如何制作一个后退按钮？

如需创建后退按钮，请使用 data-rel="back" 属性（这会忽略锚的 href 值）：

```
<a href="#" data-role="button" data-rel="back">返回</a>
```

2. 如何在面板上添加主题样式 b？

在主题上添加主题样式的方法比较简单，代码如下：

```
<div data-role="panel" id="myPanel" data-theme="b">
```

面板添加主题样式 b 后的效果如图 18-51 所示。



图 18-51 主题添加主题后的效果

第19章　jQuery Mobile事件

页面有了事件就有了“灵魂”，可见事件对于页面是多么重要，这是因为事件使页面具有了动态性和响应性，如果没有事件将很难完成页面与用户之间的交互。jQuery Mobile 针对移动端提供了各种浏览器事件，包括页面事件、触摸事件、滑动事件、定位事件等。本章就来介绍如何使用 jQuery Mobile 的事件。

19.1　页面事件

jQuery Mobile 针对各个页面生命周期的事件可以分为以下几种。

（1）初始化事件：分别在页面初始化之前，页面创建时和页面初始化之后触发的事件。

（2）外部页面加载事件：外部页面加载时触发事件。

（3）页面过渡事件：页面过渡时触发事件。

使用 jQuery Mobile 事件的方法比较简单，只需要使用 on()方法指定要触发的时间并设置事件处理函数即可，语法格式如下：

```
$(document).on(事件名称,选择器,事件处理函数)
```

其中选择器可选参数，如果省略该参数，表示事件应用于整个页面而不限定哪一个组件。

19.1.1　初始化事件

初始化事件发生的时间包括页面初始化之前、页面创建时和页面创建后。下面将详细介绍初始化事件。

1. Mobileinit

当 jQuery Mobile 开始执行时，首先会触发 mobileinit 事件。如果想更改 jQuery Mobile 的默认值时，就可以将函数绑定到 mobileinit 事件。语法格式如下：

```
$(document).on("mobileinit",function(){
   // jQuery 事件
```

```
});
例如jQuery Mobile开始执行任何操作时都会使用Ajax的方式，如果不想使用Ajax，可以在
mobileinit事件中将$.mobile.ajaxEnabled更改为false，代码如下：
$(document).on("mobileinit",function(){
  $.mobile.ajaxEnabled=false;
});
```

这里需要注意的是，上面的代码要放在引用 jquery.mobile.js 之前。

2. jQuery Mobile Initialization 事件

jQuery Mobile Initialization 事件主要包括 pagebeforecreate 事件、pagecreate 事件和 pageinit 事件，它们的区别如下：

（1）pagebeforecreate 事件：发生在页面 DOM 加载后，正在初始化时，语法格式如下：

```
$(document).on("pagebeforecreate",function(){
   // 程序语句
});
```

（2）pagecreate 事件：发生在页面 DOM 加载完成，初始化也完成时，语法格式如下：

```
$(document).on("pagecreate",function(){
   // 程序语句
});
```

（3）pageinit 事件：发生在页面初始化完成以后，语法格式如下：

```
$(document).on("pageinit",function(){
   // 程序语句
});
```

下面通过一个综合案例来学习上面三个事件触发的时机。

【例 19.1】（实例文件：ch19\19.1.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
<script>
$(document).on("pagebeforecreate",function(){
   alert("注意，pagebeforecreate事件开始触发");
});
$(document).on("pagecreate",function(){
  alert("注意，pagecreate事件触发开始触发");
```

```
});
$(document).on("pageinit",function(){
  alert("注意，pageinit事件开始触发");
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>几回花下坐吹箫，银汉红墙入望遥。</p>
    <a href="#second">下一页</a>
  </div>
  <div data-role="footer">
    <h1>清代诗人</h1>
  </div>
</div>
<div data-role="page" id="second">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>似此星辰非昨夜，为谁风露立中宵。</p>
    <a href="#first">上一页</a>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

模拟器中预览程序的效果，各个事件的执行顺序如图 19-1 所示。单击三次【确定】按钮后，结果如图 19-2 所示。

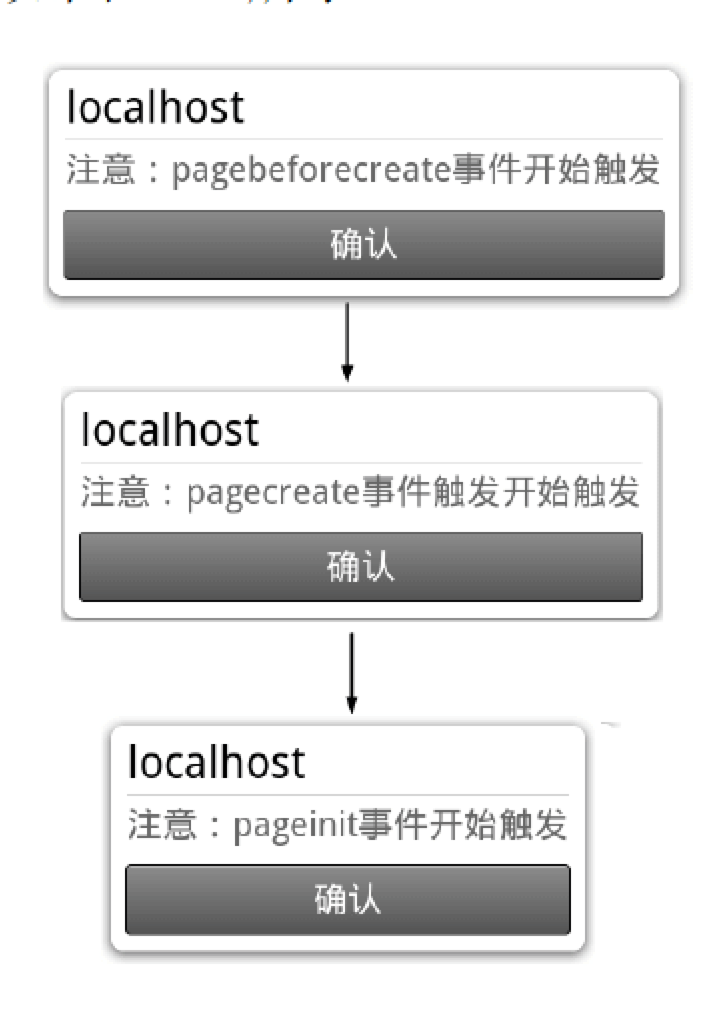


图 19-1　初始化事件



图 19-2　页面最终效果

19.1.2 外部页面加载事件

外面页面加载时，最常见的加载事件如下：

1. pagebeforeload 事件

pagebeforeload 事件在外部页面加载前触发，语法格式如下：

```
<script>
$(document).on("pagebeforeload",function(){
  alert("有外部文件将被加载);
});
</script>
```

2. pageload 事件

当页面加载成功时，触发 pageload 事件。语法格式如下：

```
<script>
$(document).on("pageload",function(event,data){
  alert("pageload事件触发!\nURL: " + data.url);
});
</script>
```

pageload 事件的函数的参数含义如下：

（1）event：任何 jQuery 的事件属性，如 event.type、event.pageX 和 target 等。

（2）data：包含以下属性。

- url：页面的 url 地址，是字符串类型。
- absUrl：绝对地址，是字符串类型。
- dataUrl：地址栏 URL，是字符串类型。
- options：$.mobile.loadPage()指定的选项，是对象类型。
- xhr:XMLHttpRequest 对象，是对象类型。
- textStatus:对象状态或空值，返回状态。

3. pageloadfailed 事件

如果页面载入失败，就触发 pageloadfailed 事件。默认地，将显示 "Error Loading Page" 消息。语法格式如下：

```
$(document).on("pageloadfailed",function(event,data){
  alert("抱歉，被请求页面不存在。");
});
</script>
```

下面通过一个例子来理解上述事件触发时机。

【例 19.2】（实例文件：ch19\19.2.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
<script>
$(document).on("pageload",function(event,data){
alert("pageload事件触发!\nURL: " + data.url);
});
$(document).on("pageloadfailed",function(){
  alert("抱歉，被请求页面不存在。");
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
    <p>众鸟高飞尽，孤云独去闲。相看两不厌，只有敬亭山。</p>
    <a href="123.html">下一页</a>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

模拟器中预览如图 19-3 所示。单击【下一页】按钮，结果如图 19-4 所示。

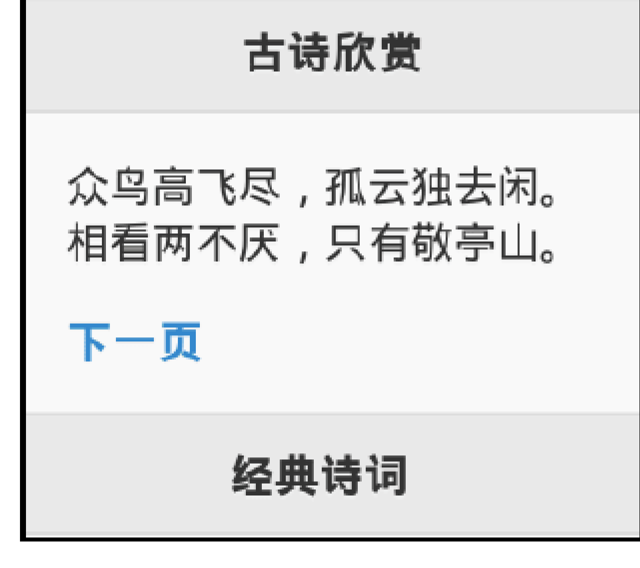


图 19-3　程序预览效果



图 19-4　触发 pageloadfailed 事件

19.1.3 页面过渡事件

在 jQuery Mobile 中，在当前页面过渡到下一页时，会触发以下几个事件。

（1）pagebeforeshow 事件：在当前页面触发，在过渡动画开始前。

（2）pageshow 事件：在当前页面触发，在过渡动画完成后。

（3）pagebeforehide 事件：在下一页触发，在过渡动画开始前。

（4）pagehide 事件：在下一页触发，过渡动画完成后。

下面通过一个案例来学习页面过渡事件的触发时机。

【例 19.3】（实例文件：ch19\19.3.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
    <script>
    $(document).on("pagebeforeshow","#second",function(){
      alert("触发 pagebeforeshow 事件，下一页即将显示");
    });
    $(document).on("pageshow","#second",function(){
      alert("触发 pageshow 事件，现在显示下一页");
    });
    $(document).on("pagebeforehide","#second",function(){
      alert("触发 pagebeforehide 事件，下一页即将隐藏");
    });
    $(document).on("pagehide","#second",function(){
      alert("触发 pagehide 事件，现在隐藏下一页");
    });</script>
    </head>
    <body>
    <div data-role="page" id="first">
      <div data-role="header">
        <h1>古诗欣赏</h1>
      </div>
      <div data-role="content" class="content">
        <p>众鸟高飞尽，孤云独去闲。相看两不厌，只有敬亭山。</p>
        <a href="#second">下一页</a>
      </div>
      <div data-role="footer">
        <h1>经典诗词</h1>
      </div>
```

```
    </div>

  <div data-role="page" id="second">
    <div data-role="header">
      <h1>古诗欣赏</h1>
    </div>
    <div data-role="content" class="content">
      <p>众鸟高飞尽，孤云独去闲。相看两不厌，只有敬亭山。</p>
      <a href="#first">上一页</a>
    </div>
    <div data-role="footer">
      <h1>经典诗词</h1>
    </div>
  </div>
  </body>
  </html>
```

模拟器中预览如图 19-5 所示。单击【下一页】按钮，事件触发顺序如图 19-6 所示。

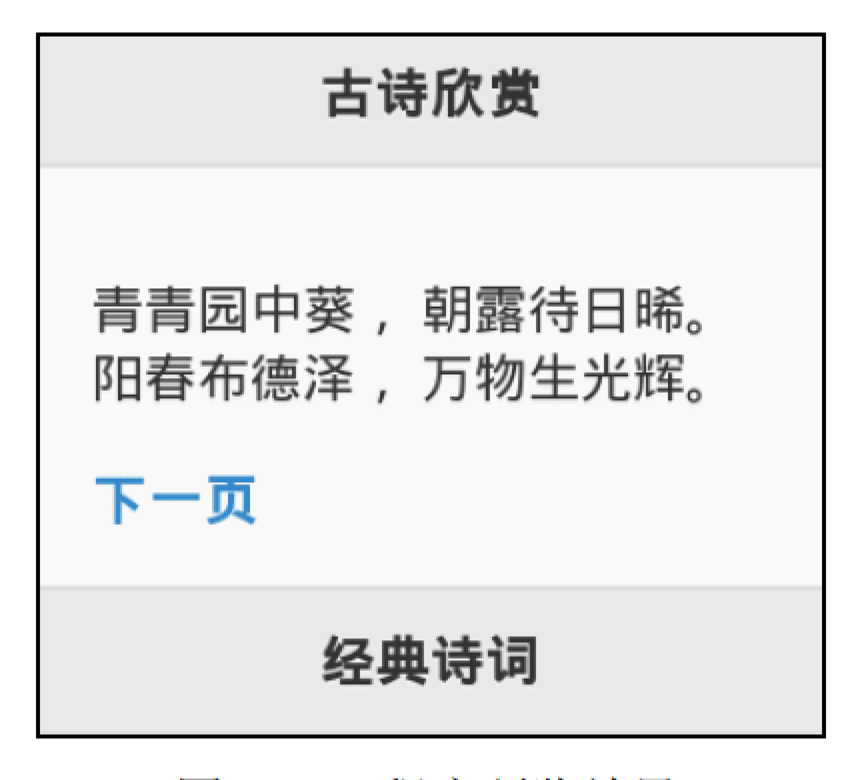


图 19-5　程序预览效果



图 19-6　当前页面触发事件顺序

单击【确定】按钮，进入下一页中，如图 19-7 所示。单击【上一页】按钮，事件触发顺序如图 19-8 所示。

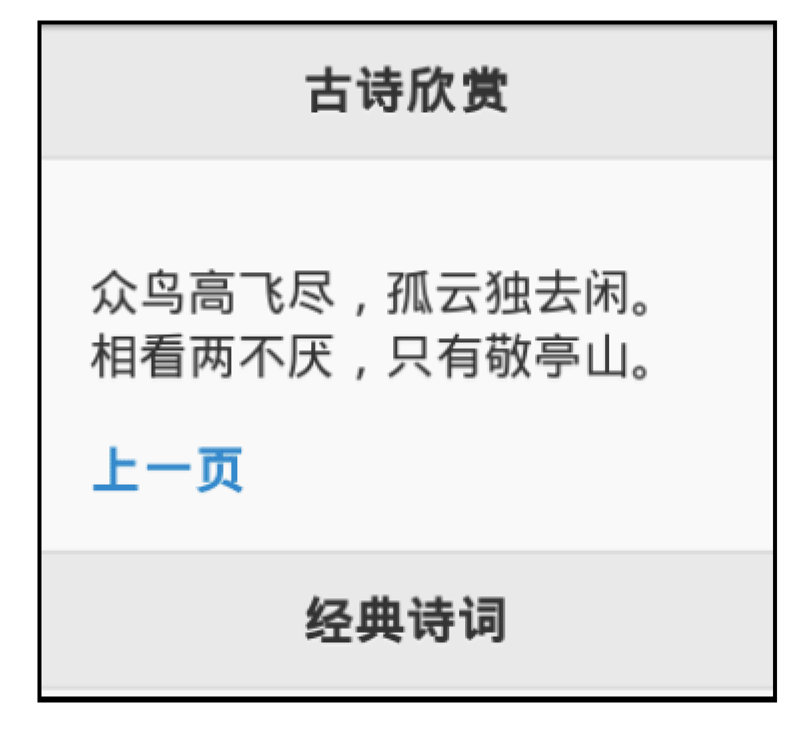


图 19-7　程序预览效果



图 19-8　下一页触发事件顺序

19.2 触摸事件

针对移动端浏览器提供了触摸事件，表示当用户触摸屏幕时触发的事件，包括点击事件和滑动事件。

19.2.1 点击事件

点击事件包括 tap 事件和 taphold 事件，下面将详细介绍它们的用法和区别。

1. tap 事件

当用户点击页面上的元素时，会触发点击(tap)事件，语法如下：

```
$("p").on("tap",function(){
  $(this).hide();
});
```

上面代码作用是点击 p 组件后，将会将该组件隐藏。

下面通过一个案例来讲解点击事件的使用方法。

【例 19.4】（实例文件：ch19\19.4.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
<script>
$("div").on("tap",function(){
  $(this).css("color","green");
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
        <p>黄师塔前江水东，春光懒困倚微风。桃花一簇开无主，可爱深红爱浅红。</p>
  </div>
  <div data-role="footer">
```

```
        <h1>经典诗词</h1>
      </div>
    </div>
    </body>
    </html>
```

模拟器中预览如图 19-9 示。在页面中诗词上面点击，即可发现 div 块内文字的颜色变成了绿色，如图 19-10。

古诗欣赏

黄师塔前江水东，春光懒困倚微风。
桃花一簇开无主，可爱深红爱浅红。

经典诗词

图 19-9　程序预览效果

古诗欣赏

黄师塔前江水东，春光懒困倚微风。
桃花一簇开无主，可爱深红爱浅红。

经典诗词

图 19-10　触发 tap 事件

2. taphold

如果点击页面并按住不放，则会触发 taphold 事件，语法如下：

```
    $("p").on("taphold",function(){
      $(this).hide();
    });
```

默认情况下，按住不放 750ms 之后触发 taphold 事件。用户也可以修改这个时间的长短，语法如下：

```
    $(document).on("mobileinit",function(){
      $.event.special.tap.tapholdThreshold=5000;
    });
```

修改后需要按住 5 秒以后才会触发 taphold 事件。

【例 19.5】（实例文件：ch19\19.5.html）

```
    <!DOCTYPE html>
    <html>
    <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
```

```
<script>
$(document).on("mobileinit",function(){
  $.event.special.tap.tapholdThreshold=1000
});
$(function(){
  $("img").on("taphold",function(){
   $(this).hide();
});
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>可爱宠物</h1>
  </div>
  <div data-role="content" class="content">
<img src=19.1.jpg > <br>
        <p>按住图片1秒后隐藏图片哦！</p>
  </div>
  <div data-role="footer">
    <h1>动物天地</h1>
  </div>
</div>
</body>
</html>
```

模拟器中预览如图 19-11 所示。点击图片 1 秒后，即可发现图片被隐藏了，如图 19-12 所示。



图 19-11　程序预览效果

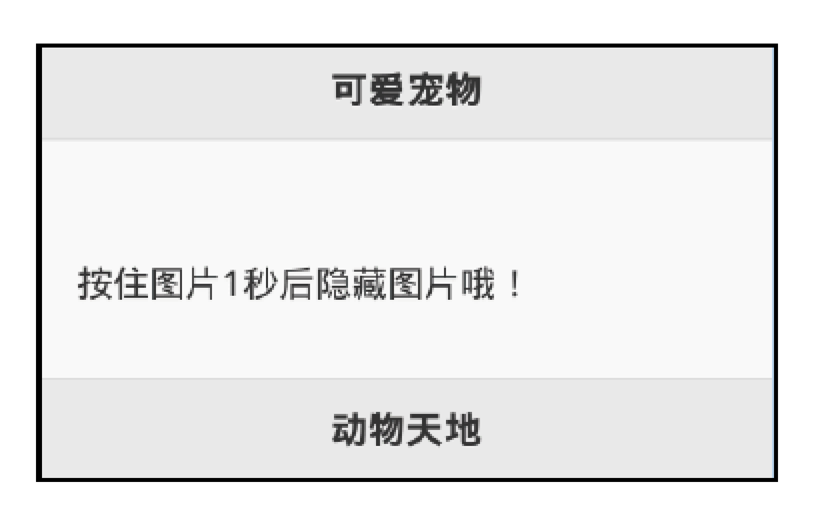


图 19-12　触发 taphold 事件

19.2.2　滑动事件

滑动事件是在用户一秒内水平拖拽大于 30 像素，或者纵向拖曳小于 20 像素的事件发生时触发的事件。滑动事件使用 swipe 语法来捕捉，语法如下：

```
$("p").on("swipe",function(){
  $("span").text("滑动检测!");
});
```

上述语法是捕捉 p 组件的滑动事件，并将消息显示在 span 组件中。

向左滑动事件会在用户向左拖动元素大于 30 像素时触发，使用 swipeleft 语法来捕捉，语法如下：

```
$("p").on("swipeleft",function(){
  $("span").text("向左滑动检测!");
});
```

向右滑动事件在用户向右拖动元素大于 30 像素时触发，使用 swiperight 语法来捕捉，语法如下：

```
$("p").on("swiperight,function(){
  $("span").text("向右滑动检测!");
});
```

下面以向右滑动事件为例进行讲解。

【例 19.6】（实例文件：ch19\19.6.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
<script>
$(document).on("pagecreate","#first",function(){
   $("img").on("swiperight",function(){
    alert("干嘛向右滑动我!!");
   });
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
     <h1>可爱宠物</h1>
  </div>
  <div data-role="content" class="content">
```

```
<img src=19.2.jpg > <br>
      <p>向右滑动图片查看效果</p>
  </div>
  <div data-role="footer">
     <h1>动物天地</h1>
  </div>
</div>
</body>
</html>
```

模拟器中预览如图 19-13 所示。向右滑动图片，效果如图 19-14 所示。

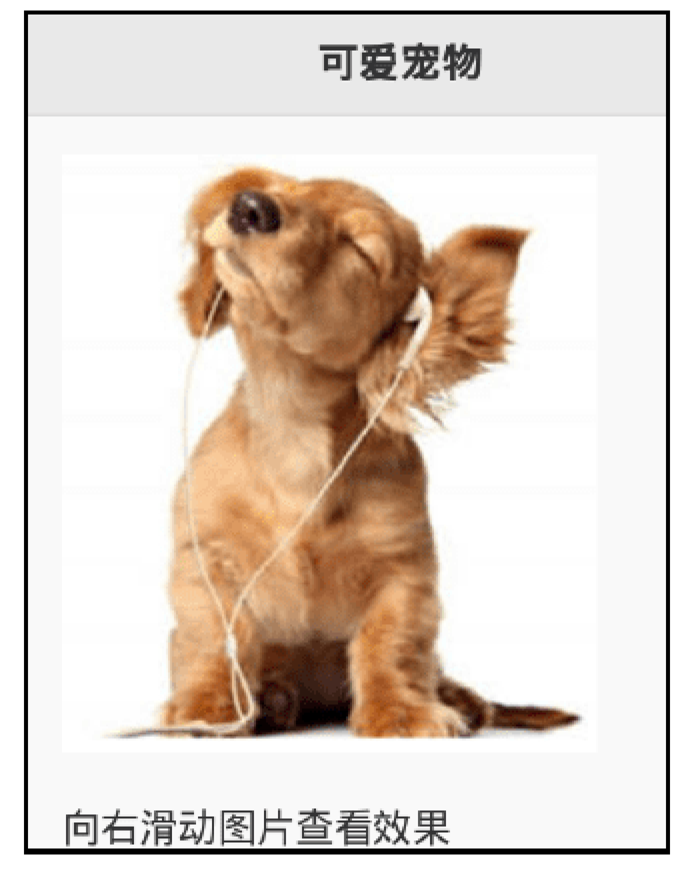


图 19-13　程序预览效果

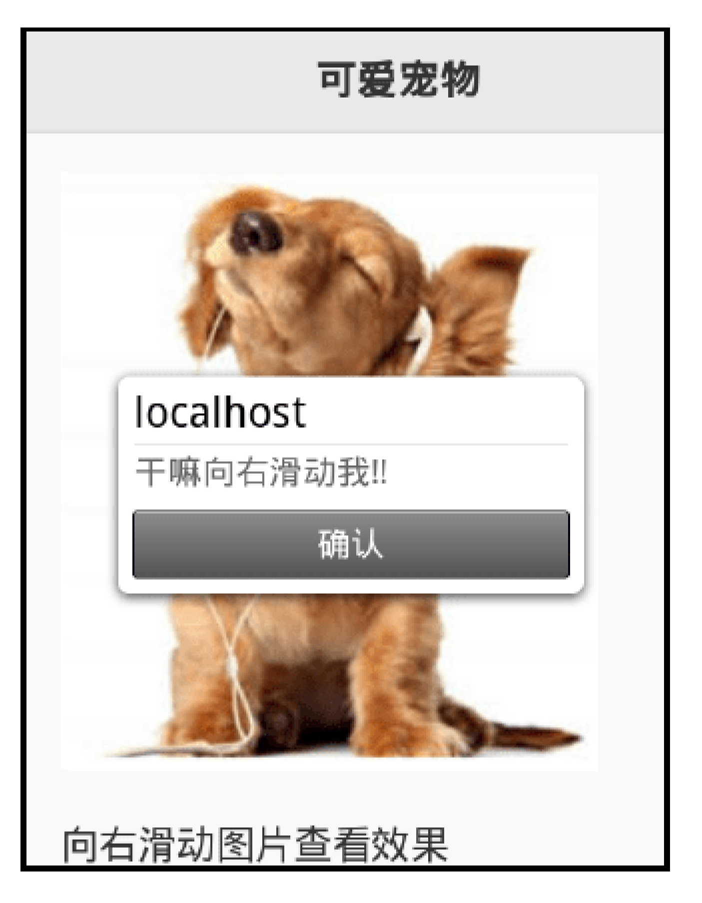


图 19-14　触发向右滑动事件

19.3　滚屏事件

jQuery Mobile 提供了两种滚屏事件，分别是滚屏开始时触发 Scrollstart 事件和滚动结束时触发 Scrollstop 事件。

1. Scrollstart 事件

scrollstart 事件是在用户开始滚动页面时触发。语法如下：

```
  $(document).on("scrollstart",function(){
    alert("屏幕开始滚动了!");
  });
```

下面通过一个案例来理解 Scrollstart 事件。

【例 19.7】（实例文件：ch19\19.7.html）

```
<!DOCTYPE html>
<html>
<head>
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
     <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
    <script>
    $(document).on("pagecreate","#first",function(){
      $(document).on("scrollstart",function(){
        alert("屏幕开始滚动了！");
      });
    });
    </script>
    </head>
    <body>
    <div data-role="page" id="first">
      <div data-role="header">
        <h1>古诗欣赏</h1>
      </div>
      <div data-role="content" class="content">
    <p>西施越溪女，出自苎萝山。</p>
    <p>秀色掩今古，荷花羞玉颜。</p>
    <p>浣纱弄碧水，自与清波闲。</p>
    <p>皓齿信难开，沉吟碧云间。</p>
    <p>勾践徵绝艳，扬蛾入吴关。</p>
    <p>提携馆娃宫，杳渺讵可攀。</p>
    <p>一破夫差国，千秋竟不还。</p>
    <p>西施越溪女，出自苎萝山。</p>
    <p>秀色掩今古，荷花羞玉颜。</p>
    <p>浣纱弄碧水，自与清波闲。</p>
    <p>皓齿信难开，沉吟碧云间。</p>
    <p>勾践徵绝艳，扬蛾入吴关。</p>
    <p>提携馆娃宫，杳渺讵可攀。</p>
    <p>一破夫差国，千秋竟不还。</p>
      </div>
      <div data-role="footer">
        <h1>经典诗词</h1>
      </div>
    </div>
    </body>
    </html>
```

模拟器中预览如图 19-15 所示。向上滚动屏幕，效果如图 19-16 所示。



图 19-15　程序预览效果



图 19-16　触发滚屏事件

2. Scrollstop 事件

scrollstop 事件是在用户停止滚动页面时触发，语法如下：

```
$(document).on("scrollstop",function(){
 alert("停止滚动!");
});
```

下面通过一个案例来理解 Scrollstart 事件。

【例 19.8】（实例文件：ch19\19.8.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
<script>
$(document).on("pagecreate","#first",function(){
  $(document).on("scrollstop",function(){
    alert("屏幕已经停止滚动了!");
  });
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
```

```
    <div data-role="content" class="content">
  <p>噫吁嚱，危乎高哉！</p>
  <p>蜀道之难，难于上青天！</p>
  <p>蚕丛及鱼凫，开国何茫然！</p>
  <p>尔来四万八千岁，不与秦塞通人烟。</p>
  <p>西当太白有鸟道，可以横绝峨嵋巅。</p>
  <p>地崩山摧壮士死，然后天梯石栈方钩连。</p>
  <p>上有六龙回日之高标，下有冲波逆折之回川。</p>
  <p>黄鹤之飞尚不得过，猿猱欲度愁攀援。</p>
  <p>青泥何盘盘，百步九折萦岩峦。</p>
  <p>扪参历井仰胁息，以手抚膺坐长叹。</p>
  <p>问君西游何时还？畏途巉岩不可攀。</p>
  <p>但见悲鸟号古木，雄飞从雌绕林间。</p>
  <p>又闻子规啼夜月，愁空山。</p>
  <p>蜀道之难，难于上青天，使人听此凋朱颜。</p>
  <p>连峰去天不盈尺，枯松倒挂倚绝壁。</p>
  <p>飞湍瀑流争喧豗，砯崖转石万壑雷。</p>
  <p>其险也若此，嗟尔远道之人，胡为乎来哉。</p>
  <p>剑阁峥嵘而崔嵬，一夫当关，万夫莫开。</p>
  <p>所守或匪亲，化为狼与豺。</p>
  <p>朝避猛虎，夕避长蛇，磨牙吮血，杀人如麻。</p>
  <p>锦城虽云乐，不如早还家。</p>
  <p>蜀道之难，难于上青天，侧身西望长咨嗟。</p>
    </div>
    <div data-role="footer">
      <h1>经典诗词</h1>
    </div>
  </div>
  </body>
  </html>
```

模拟器中预览如图 19-17 所示。向上滚动屏幕，停止后效果如图 19-18 所示。

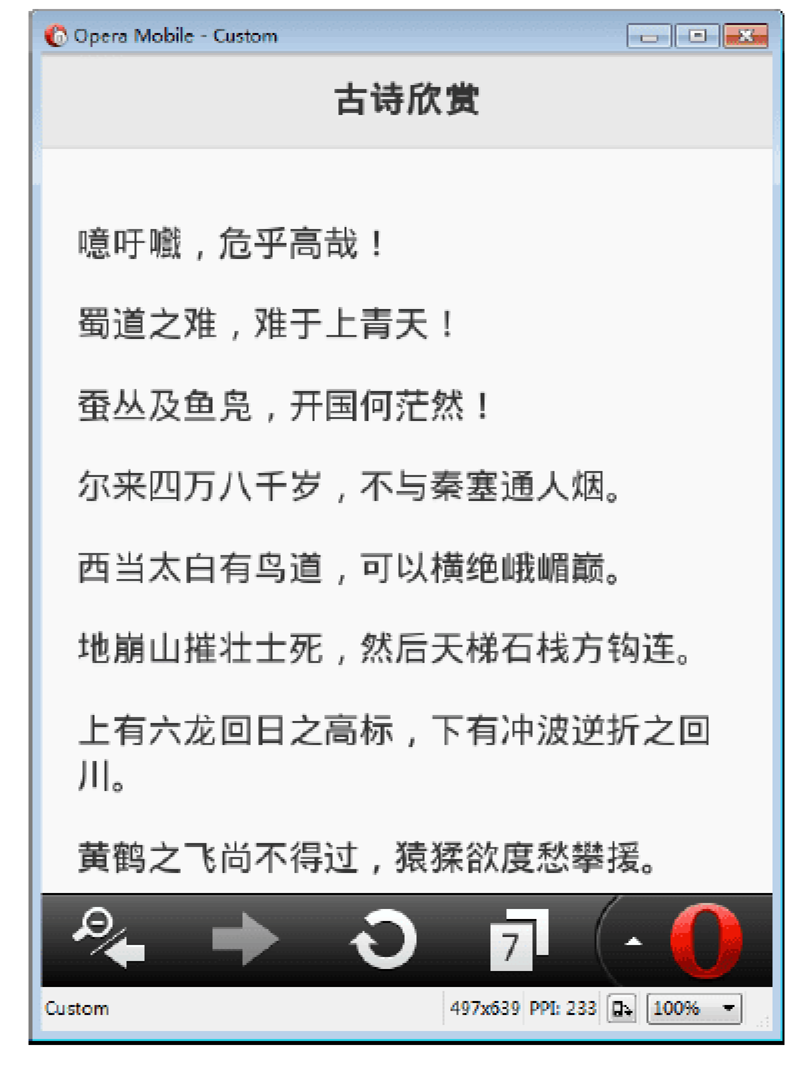


图 19-17　程序预览效果



图 19-18　触发滚屏事件

19.4 定位事件

当移动设备水平或垂直翻转时触发定位事件。也就是常说的方向改变（orientationchange）事件。

在使用定位事件时，请将 orientationchange 事件绑定到 window 对象上，语法如下：

```
$(window).on("orientationchange",function(event){
alert("设备的方向改变为"+ event.orientation);
});
```

这里的 event 对象用来接收 orientation 属性值，用 event.orientation 返回的是设备是水平还是垂直，类型为字符串，如果是横行，返回值为 landscape，如果是纵向，返回值为 portrait。

下面通过一个案例来理解 orientationchange 事件。

【例 19.9】（实例文件：ch19\19.9.html）

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></scri
pt>
<script type="text/javascript">
    $(document).on("pageinit",function(event){
        $( window ).on( "orientationchange", function( event ) {
         if(event.orientation == "landscape")
           $( "#orientation" ).text( "现在是水平模
式!" ).css({"background-color":"yellow","font-size":"300%"});
         if(event.orientation == "portrait")
            $( "#orientation" ).text( "现在是垂直模
式!" ).css({"background-color":"green","font-size":"200%"});
         });
        })
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
<span id="orientation"></span><br>
<p>燕草如碧丝，秦桑低绿枝。当君怀归日，是妾断肠时。春风不相识，何事入罗帏</p>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
```

```
    </div>
    </body>
    </html>
```

模拟器中预览如图 19-19 所示。单击模拟器上的方向改变按钮，此时方向改变为水平方向，效果如图 19-20 所示。

再次单击模拟器上的方向改变按钮，此时方向改变为垂直方向，效果如图 19-21 所示。



图 19-19　程序预览效果



图 19-20　设备水平方向



图 19-21　设备垂直方向

19.5　专家解惑

绑定事件的方法 on()和 one()的区别？

绑定事件的方法 on()和 one()的作用相似，它们唯一的区别在于 one()只能执行一次。

例如当在按钮上绑定单击鼠标事件时，on()方法的程序如下：

```
<script>
$(document).on('click',function(){
   alert("这是使用on()方法绑定的事件")
});
</script>
```

小白：如何在设备方向改变时获取移动设备的高度和宽度？

大神：如果设备方向改变时要获取移动设备的长度和宽度，可以绑定 resize 事件。该事件在页面大小改变时将被触发，语法如下：

```
$(window).on("resize",function(){
  var win= $(this);    //this指的是window
alert("宽度为"+win.width()+"高度为"+ win.height());
});
```

第20章　使用最新Bootstrap 4框架

Bootstrap 是一款用于快速开发 Web 应用程序和网站的前端框架，它是基于 HTML、CSS 和 JavaScript 等技术开发的。本章将简单介绍 Bootstrap 的基本使用。

20.1　Bootstrap 概述

Bootstrap 是由 Twitter 公司主导设计研发的，基于 HTML、CSS、JavaScript 开发的简洁、直观的前端开发框架，使得 Web 开发更加快捷。Bootstrap 推出后颇受欢迎，一直是 GitHub 上的热门开源项目，可以说是目前最受欢迎的前端框架之一。

20.1.1　Bootstrap 特色

Bootstrap 是当前比较流行的前端框架，起源于 Twitter，是 Web 开发人员的一个重要工具。

1. 跨设备，跨浏览器

可以兼容所有现代主流浏览器，Bootstrap 3 不兼容 IE7 及其以下的版本，Bootstrap 4 不再支持 IE8。自 Bootstrap 3 起，框架包含了贯穿于整个库的移动设备优先的样式，重点支持各种平板电脑和智能手机等移动设备。

2. 响应布局

从 Bootstrap 2 开始，便支持响应式布局，能够自适应于台式机、平板电脑和手机，从而提供一致的用户体验。

3. 列网格布局

Bootstrap 提供了一套响应式、移动设备优先的网格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多 12 列，也可以根据自己的需要定义列数。

4. 比较全面的组件

Bootstrap 提供了实用性很强的组件，如导航、按钮、下拉菜单、表单、列表、输入框等，供开发者使用。

5. 内置 jQuery 插件

Bootstrap 提供了很多实用性的 JQuery 插件，如模态框、旋转木马等，这些插件方便开发者实现 Web 中各种常规特效。

6. 支持 HTML5 和 CSS3

因为 Bootstrap 的使用要求在 HTML5 文档类型的基础上，所以支持 HTML5 标签和语法。Bootstrap 支持 CSS3 的属性和标准，并不断完善。

7. 容易上手

只要具备 HTML 和 CSS 的基础知识，就可以开始学习 Bootstrap 并使用它。

8. 开源的代码

Bootstrap 是完全开源的，不管是个人还是企业都可以免费使用。Bootstrap 全部托管于 GitHub，并借助 GitHub 平台实现社区化的开发和共建。

20.1.2 Bootstrap 4 的更新

Bootstrap 4 相比 Bootstrap 3 有太多重要的更新，下面是其中一些更新的亮点。

（1）不再支持 IE8，使用 rem 和 em 单位：Bootstrap 4 放弃对 IE8 的支持，这意味着开发者可以放心地利用 CSS 的优点，不必再研究 CSS hack 技巧或回退机制。使用 rem 和 em 代替像素单位，更适合做响应式布局及控制组件大小。如果要支持 IE8，就只能继续用 Bootstrap 3。

（2）从 Less 到 Sass：现在，Bootstrap 已加入 Sass 的大家庭中，得益于 Libsass，Bootstrap 的编译速度比以前更快。

（3）支持选择弹性盒模型（Flexbox）：这是一项划时代的功能，即只要修改一个变量 Boolean 值，就可以让 Bootstrap 中的组件使用 Flexbox。

（4）废弃了 wells、thumbnails 和 panels，使用 cards（卡片）代替：Cards 是一个全新的概念，使用起来与 wells、thumbnails 和 panels 很像，但是更加方便。

（5）将所有 HTML 重置样式表整合到 Reboot 中：在一些用不了 Normalize.css 的地方，可以使用 Reboot 重置样式，它提供了更多选项。

（6）新的自定义选项：不再将 Flexbox、渐变、圆角、阴影等效果分放在单独的样式表中，而是将所有选项都移到一个 Sass 变量中。如果想改变默认效果，就只需要更新变量值，重新编译就可以了。

（7）重写所有 JavaScript 插件：为了利用 JavaScript 的新特性，Bootstrap 4 用 ES6 重写了所有插件。现在提供 UMD 支持、泛型拆解方法、选项类型检查等特性。

（8）更多变化：支持自定义窗体控件、空白和填充类，还包括新的实用程序类等。

20.2　下载 Bootstrap

Bootstrap 4 是 Bootstrap 的最新版本，与之前的版本相比拥有更强大的功能。Bootstrap 4 有两个版本的压缩包：一个是源代码文件，主要供学习使用；另一个是编译版，主要供直接引用。

1.下载源代码版的 Bootstrap

我们知道 Bootstrap 全部托管于 GitHub，并借助 GitHub 平台实现社区化的开发和共建，所以可以到 GitHub 上下载 Bootstrap 压缩包。使用谷歌浏览器访问 https://github.com/twbs/bootstrap/ 页面，单击“Download ZIP”按钮，下载最新版的 Bootstrap 压缩包，如图 20-1 所示。

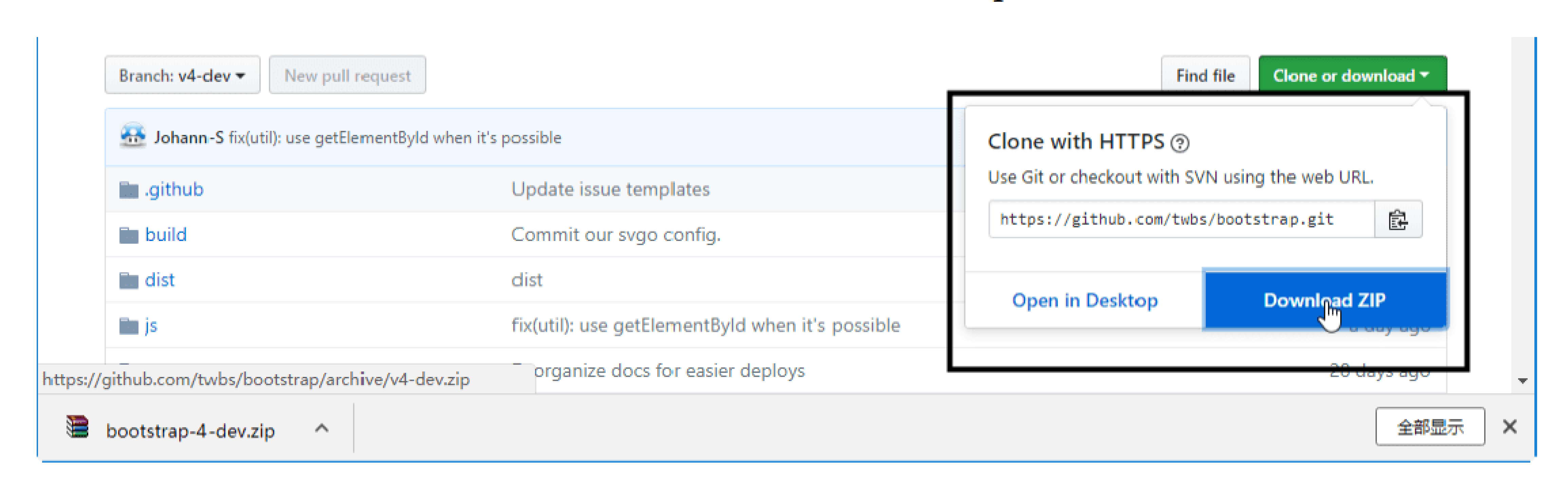


图 20-1　在 GitHub 上下载源代码文件

Bootstrap 4 源代码下载完成后解压，目录结构如图 20-2 所示。

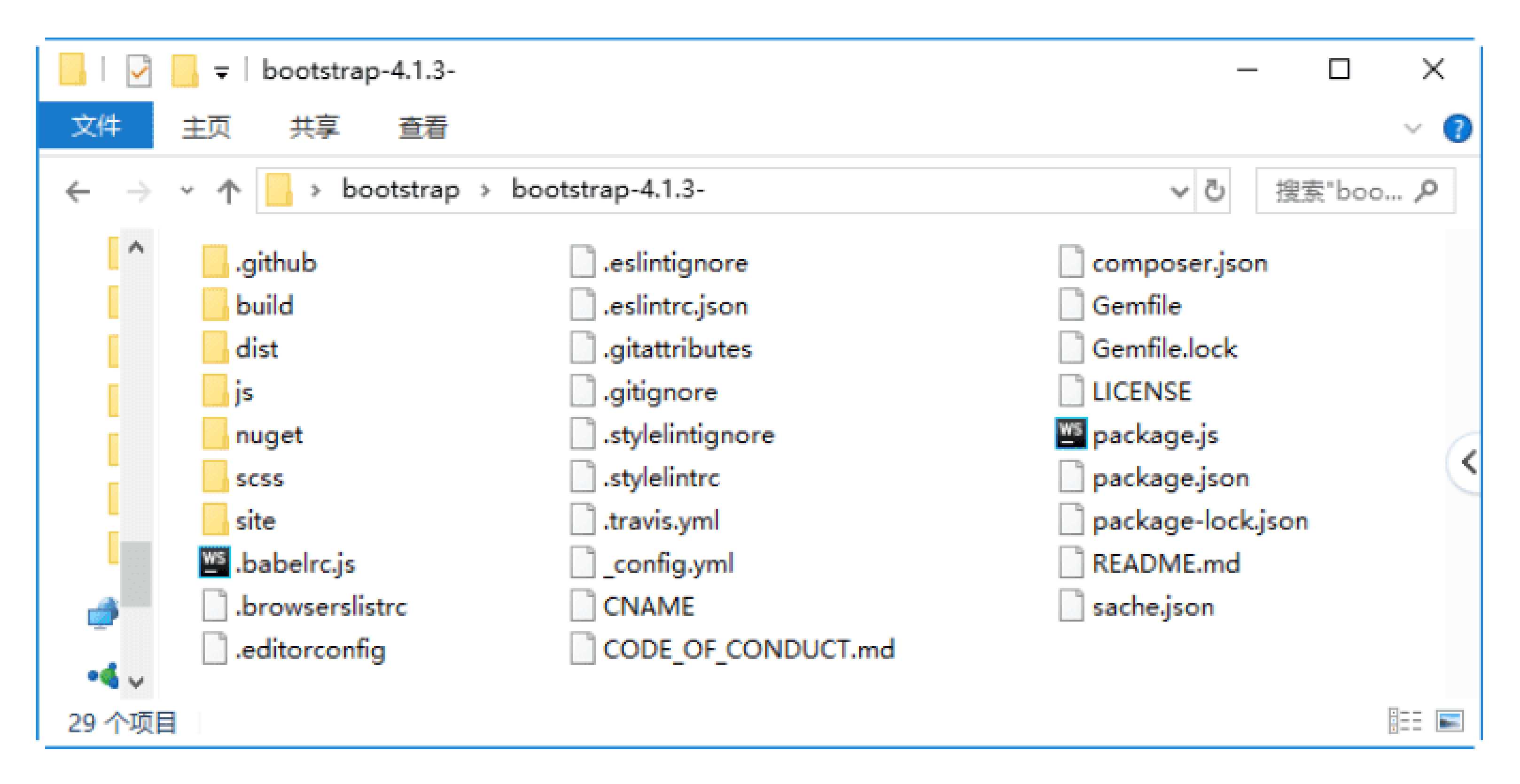


图 20-2　源代码文件的目录结构

2.下载编译版 Bootstrap

如果用户需要快速使用 Bootstrap 来开发网站，就可以直接下载经过编译、压缩后的发布版本，使用浏览器访问 http://getbootstrap.com/docs/4.1/getting-started/download/ 页面，单击

“Download”按钮，下载编译版本压缩文件，如图 20-3 所示。

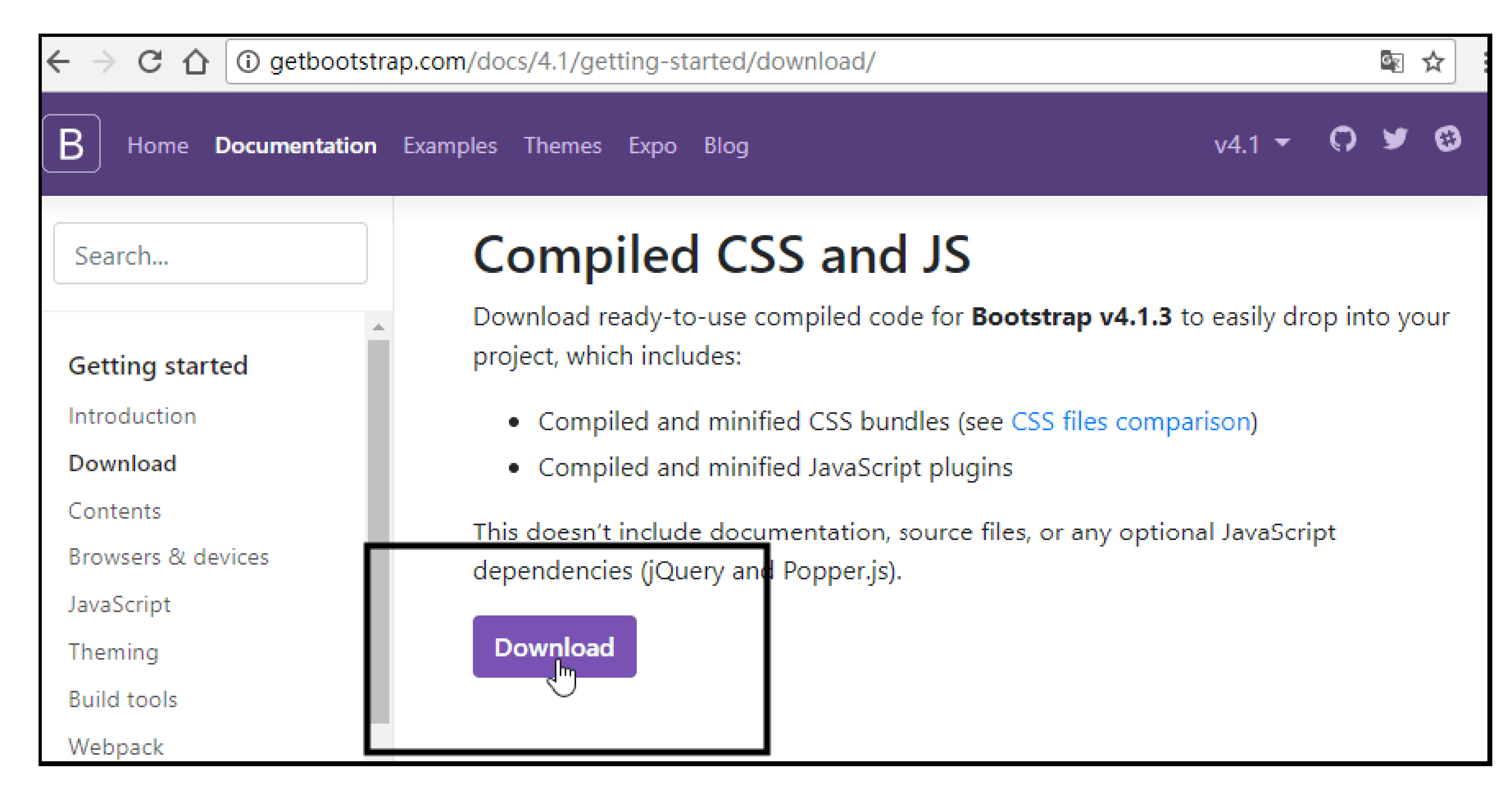


图 20-3　从官网下载编译版的 Bootstrap

编译版的压缩文件仅包含编译好的 Bootstrap 应用文件，包括 CSS 和 JS 文件，相比 Bootstrap 3 少了 fonts 字体文件，如图 20-4 所示。

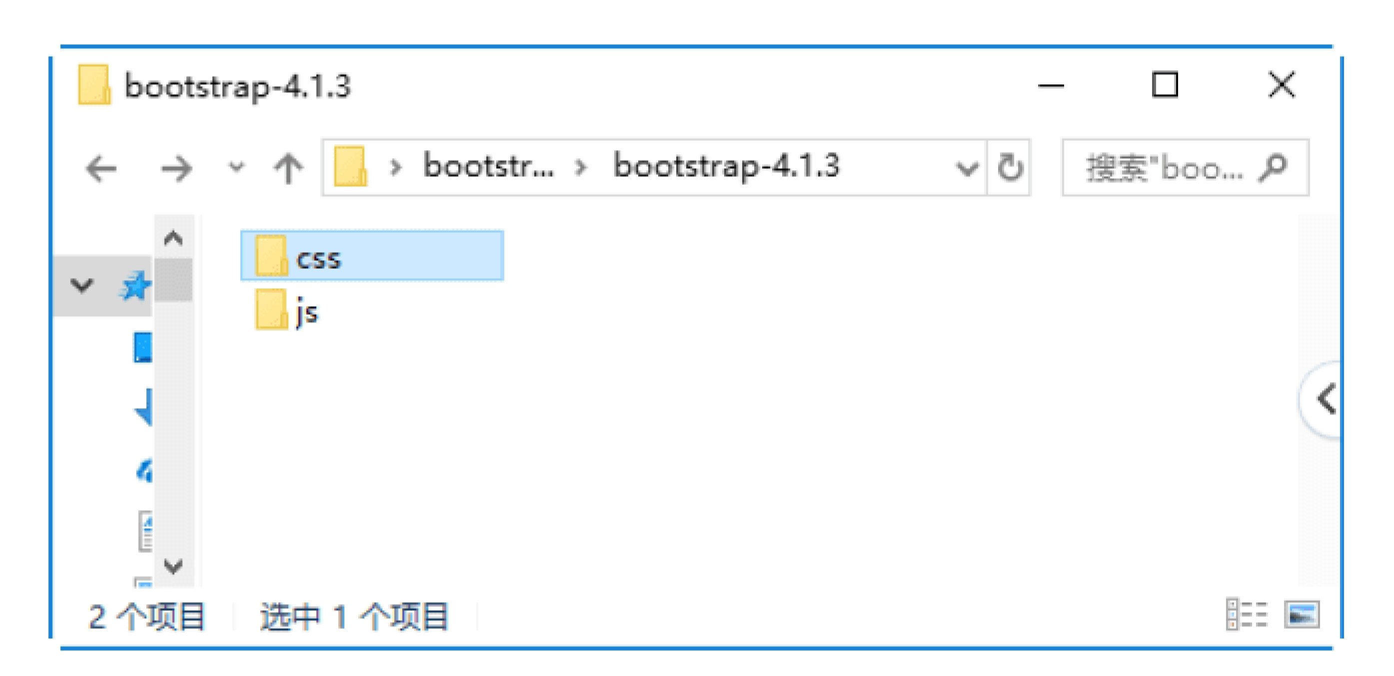


图 20-4　编译文件的目录结构

CSS 文件的目录结构如图 20-5 所示，JS 文件的目录结构如图 20-6 所示。

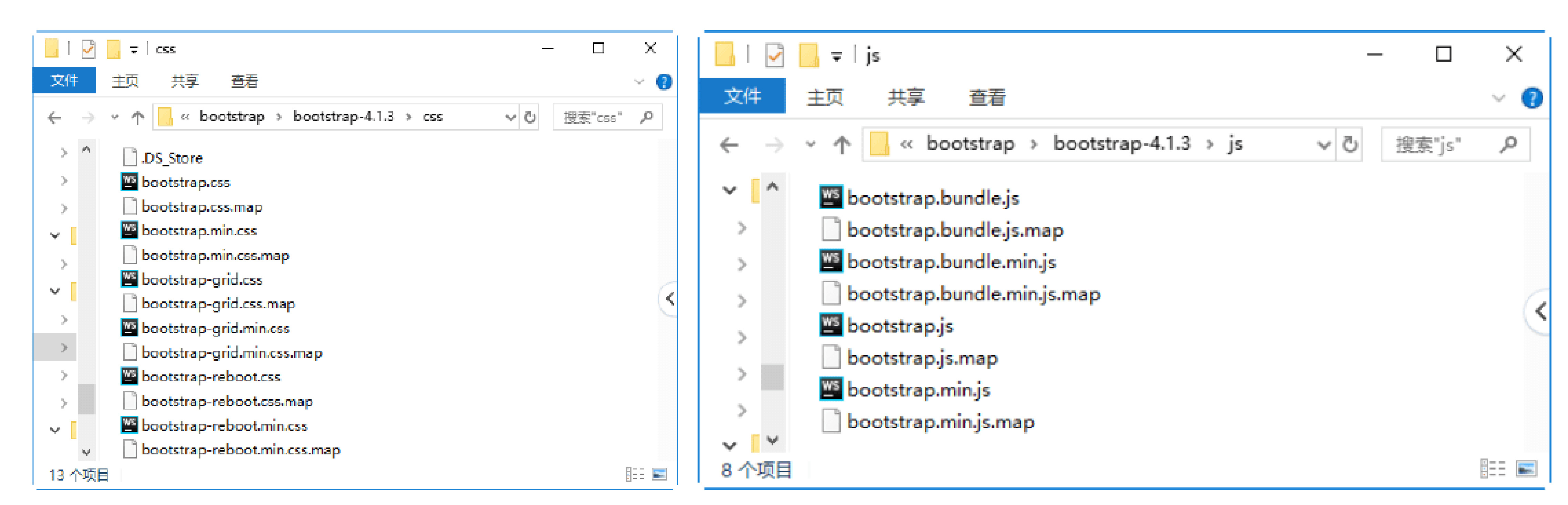


图 20-5　CSS 文件目录结构　　　　图 20-6　JS 文件目录结构

在网站目录中，导入相应的 CSS 和 JS 文件，便可以在项目中使用 Bootstrap 的效果和插件了。

20.3　安装 Bootstrap

Bootstrap 需要安装后才可以使用。

20.3.1　本地安装 Bootstrap

因为 Bootstrap 是本着移动设备优先的策略开发的，所以优先为移动设备优化代码，根据每个组件的情况并利用 CSS 媒体查询技术为组件设置合适的样式。为了确保在所有设备上正确渲染并支持触控缩放，需要将设置 viewport 属性的<meta>标签添加到<head>中，具体如下面代码所示。

```
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
```

本地安装 Bootstrap 大致可分为以下两步。

第一步：安装 Bootstrap 的基本样式，使用<link>标签引入 Bootstrap.css 样式表文件，并且放在其他样式表之前，如下代码所示。

```
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
```

第二步：调用 Bootstrap 的 JS 文件及 jQuery 框架。要注意 Bootstrap 中的许多组件需要依赖 JavaScript 才能运行，它们依赖的是 jQuery、Popper.js，Popper.js 包含在我们引入的 bootstrap.bundle.js 中。具体的引入顺序是，jQuery.js 必须放在最前面，然后是 bundle.js，最后是 Bootstrap.js，如下面的代码所示。

```
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
```

20.3.2　初次使用 Bootstrap

下面我们就来使用 Bootstrap 完成一个简单的小案例。

首先需要在页面<head>中引入 Bootstrap 核心代码文件。代码如下：

```
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
```

然后在<body>中添加一个<h1>标签，并添加 Bootstrap 中的 bg-dark 和 text-white 类。bg-dark

类用于设置<h1>标签的背景色为黑色，text-white 设置<h1>标签的字体颜色为白色，具体代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
</head>
<body>
<!--.bg-dark类用来设置背景颜色为黑色，text-white用来设置文本颜色为白色-->
<h1 class="bg-dark text-white">hello world!</h1>
</body>
</html>
```

在 IE 11.0 浏览器中显示效果如图 20-7 所示。

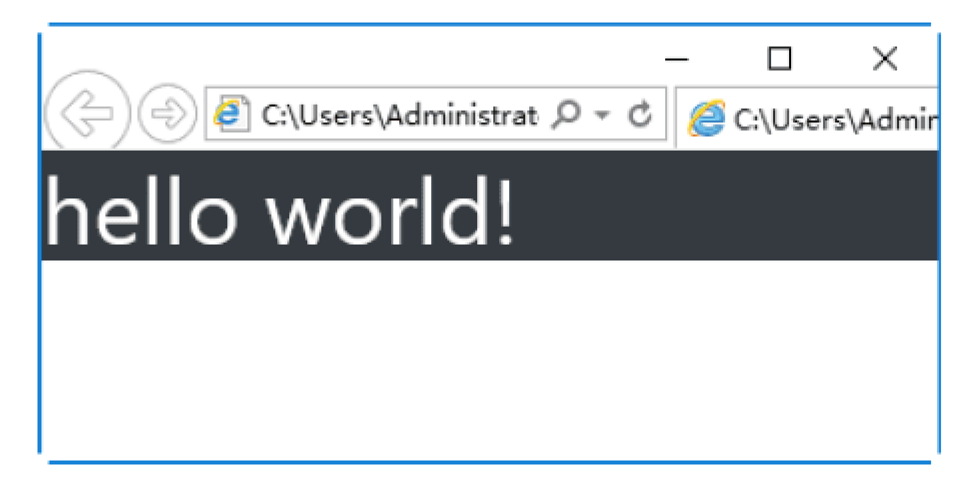


图 20-7　初始 Bootstrap

注意，在<head>中引入的核心代码在后续的内容中将省略，读者务必加上。

20.4　使用常用组件

Bootstrap 提供了大量可复用的组件，下面简单介绍其中一些常用的组件，更加详细的内容请参考官方文档。

20.4.1　使用下拉菜单

下拉菜单是网页中经常看到的效果之一，使用 Bootstrap 很容易就可以实现。

在 Bootstrap 中可以使用一个按钮或链接来打开下拉菜单，按钮或链接需要添加.dropdown-toggle 类和 data-toggle="dropdown"属性。

在菜单元素中添加.dropdown-menu 类来实现下拉，然后在下拉菜单的选项中添

加.dropdown-item 类。在下面的案例中使用一个列表来设计菜单。

【例 20.1】（实例文件：ch20\Chap20.1.html）下拉菜单。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title> </title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <div class="container">
        <div>
            <!--.btn类设置a标签为按钮，.dropdown-toggle类和data-toggle="dropdown"
属性类别，用来激活下拉菜单-->
            <a href="#" class="dropdown-toggle" data-toggle="dropdown">下拉菜单
</a>
            <!--.dropdown-menu用来指定被激活的菜单-->
            <ul class="dropdown-menu">
                <!--.dropdown-item添加列表元素的样式-->
                <li><a href="#" class="dropdown-item">新闻</a></li>
                <li><a href="#" class="dropdown-item">电视</a></li>
                <li><a href="#" class="dropdown-item">电影</a></li>
            </ul>
        </div>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-8 所示。

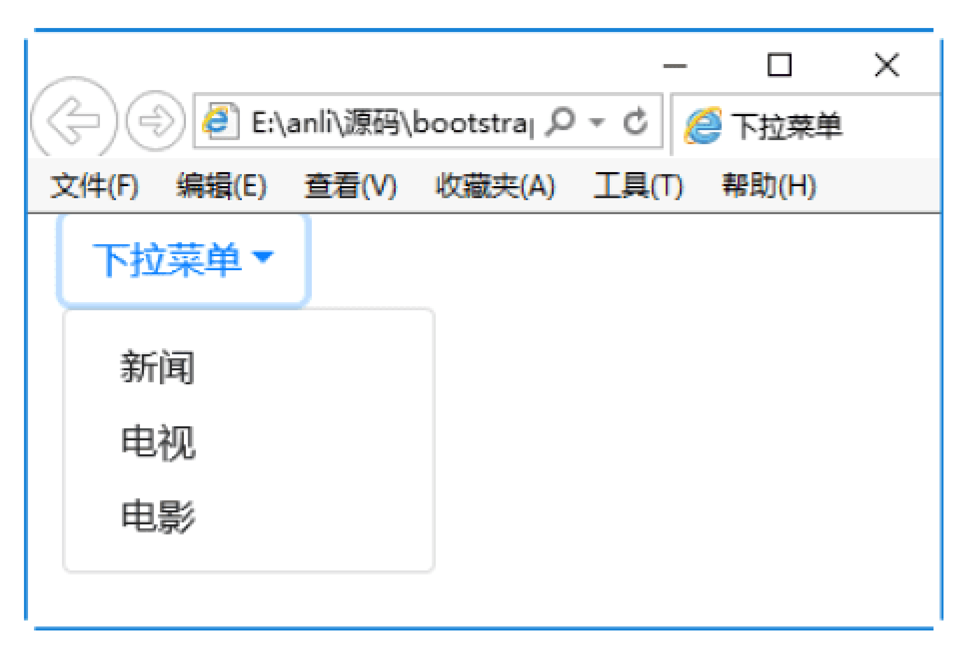


图 20-8 下拉菜单

20.4.2 使用按钮组

用含有.btn-group 类的容器把一系列含有.btn 类的按钮包裹起来，便形成了一个页面组件——按钮组。

【例 20.2】（实例文件：ch20\Chap20.2.html）按钮组。

```
<!DOCTYPE html>
<html>
<head>
<title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
</head>
<body>
<div class="container">
    <!--使用含有.btn-group类的div来包裹按钮元素-->
    <div class="btn-group">
        <!--.btn btn-primary设置按钮为浅蓝色；.btn btn-info设置为按钮深蓝色；.btn btn-success设置按钮为绿色；.btn btn-warning设置按钮为黄色；.btn btn-danger设置按钮为红色；-->
        <button class="btn btn-primary">首页</button>
        <button class="btn btn-success">新闻</button>
        <button class="btn btn-info">电视</button>
        <button class="btn btn-warning">电影</button>
        <button class="btn btn-danger">动漫</button>
    </div>
</div>
</body>
</html>
```

在 IE 11.0 浏览器中运行的结果如图 20-9 所示。

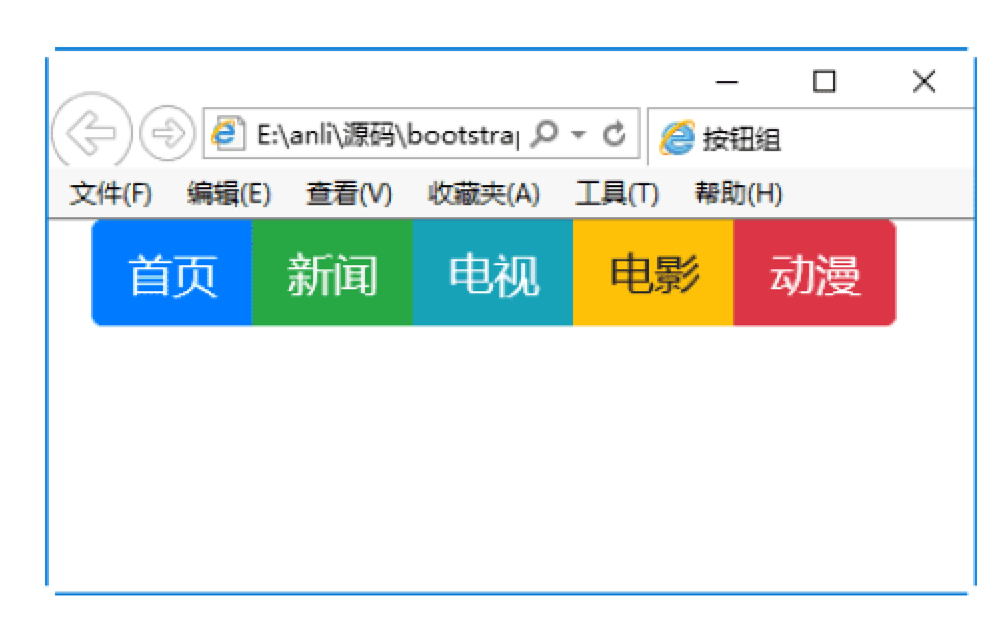


图 20-9 按钮组

20.4.3 使用导航组件

一个简单的导航栏可以通过在<ul>元素上添加.nav 类、每个<li>元素上添加.nav-item 类、每个链接上添加.nav-link 类来实现，如下面的代码所示。

【例 20.3】（实例文件：ch20\Chap20.3.html）简单导航。

```
<!DOCTYPE html>
<html>
<head>
<title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
</head>
<body>
<div class="container">
    <p>基本的导航:</p>
    <!--在ul中添加.nav类创建导航栏-->
    <ul class="nav">
        <!--在li中添加.nav-item,在a中添加.nav-link设置导航的样式-->
        <li class="nav-item"><a class="nav-link" href="#">小说</a></li>
        <li class="nav-item"><a class="nav-link" href="#">音乐</a></li>
        <li class="nav-item"><a class="nav-link" href="#">视频</a></li>
        <li class="nav-item"><a class="nav-link" href="#">游戏</a></li>
    </ul>
</div>
</body>
</html>
```

在 IE 1.0 浏览器中运行的结果如图 20-10 所示。



图 20-10 基本的导航

Bootstrap 的导航组件都是建立在基本导航之上的，可以通过扩展基础的.nav 组件来实现别样的导航样式。

1．标签页导航

在基本导航中为<ul>元素添加.nav-tabs 类，对于选中的选项使用.active 类并为每个链接添加 data-toggle="tab"属性类别，便可以实现标签页导航了。

【例 20.4】（实例文件：ch20\Chap20.4.html）标签页导航。

```
<!DOCTYPE html>
<html>
<head>
<title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
</head>
<body>
<div class="container">
    <p>标签页导航</p>
    <!--在ul中添加.nav和.nav-tabs，.nav-tabs用来设置标签页导航-->
    <ul class="nav nav-tabs">
        <!--在li中添加.nav-item，在a中添加.nav-link，对于选中的选项添加.active类
-->
        <!--添加data-toggle="tab"属性类别，是去掉a标签的默认行为，实现动态切换导航
的active属性效果-->
        <li class="nav-item"><a class="nav-link active" href="#"
data-toggle="tab">健康</a></li>
        <li class="nav-item"><a class="nav-link" href="#"
data-toggle="tab">时尚</a></li>
        <li class="nav-item"><a class="nav-link" href="#"
data-toggle="tab">减肥</a></li>
        <li class="nav-item"><a class="nav-link" href="#"
data-toggle="tab">美食</a></li>
        <li class="nav-item"><a class="nav-link" href="#"
data-toggle="tab">交友</a></li>
        <li class="nav-item"><a class="nav-link" href="#"
data-toggle="tab">社区</a></li>
    </ul>
</div>
</body>
</html>
```

在 IE 11.0 浏览器中运行的结果如图 20-11 所示。

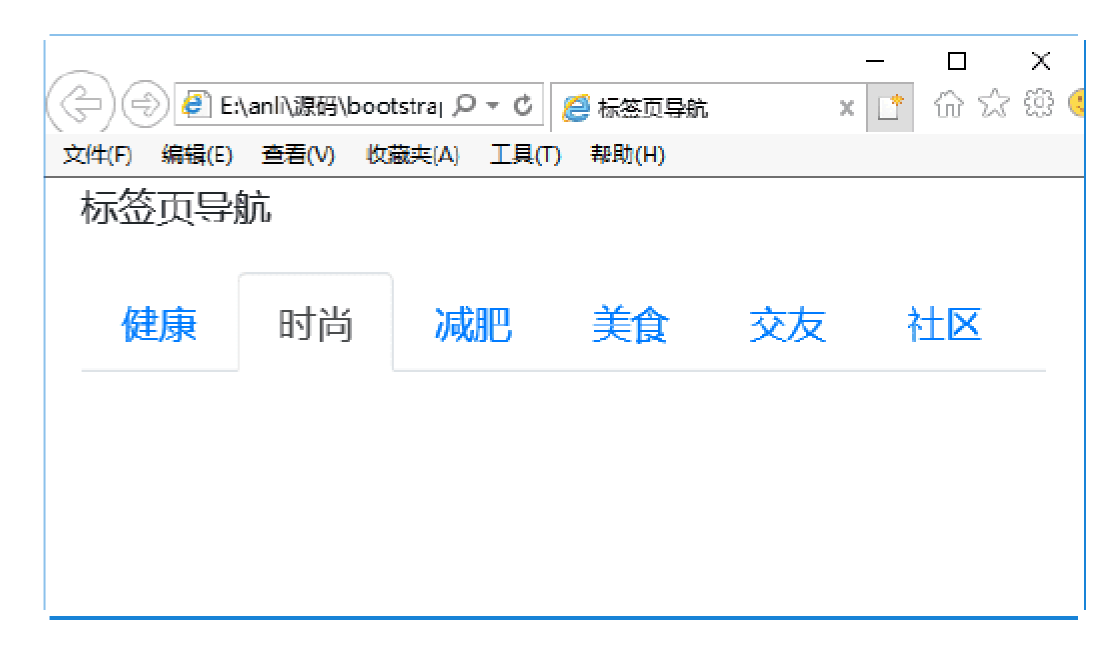


图 20-11　标签页导航

2. 胶囊导航

在基本导航中为<ul>添加.nav-pills 类，对于选中的选项使用.active 类并为每个链接添加 data-toggle="pill"属性类别，便可以实现胶囊导航了。

【例 20.5】（实例文件：ch20\Chap20.5.html）胶囊导航。

```
<!DOCTYPE html>
<html>
<head>
<title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
</head>
<body>
<div class="container">
    <p>胶囊导航</p>
    <!--在ul中添加.nav和.nav-pills，.nav-pills类用来设置胶囊导航-->
    <ul class="nav nav-pills">
        <!--在li中添加.nav-item，在a中添加.nav-link，对于选中的选项添加.active类-->
        <!--添加data-toggle="pill"属性类别，是去掉a标签的默认行为，实现动态切换导航的active属性效果-->
        <li class="nav-item"><a class="nav-link active" href="#" data-toggle="pill">健康</a></li>
        <li class="nav-item"><a class="nav-link" href="#" data-toggle="pill">时尚</a></li>
        <li class="nav-item"><a class="nav-link" href="#" data-toggle="pill">减肥</a></li>
        <li class="nav-item"><a class="nav-link" href="#" data-toggle="pill">美食</a></li>
        <li class="nav-item"><a class="nav-link" href="#" data-toggle="pill">交友</a></li>
```

```
        <li class="nav-item"><a class="nav-link" href="#"
data-toggle="pill">社区</a></li>
      </ul>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-12 所示。



图 20-12　胶囊导航

20.4.4　绑定导航和下拉菜单

在 Bootstrap 中，下拉菜单可以与页面中的其他元素绑定使用，如导航、按钮等。

标签页导航在前面介绍过，我们只需要在标签页导航选项中添加一个下拉菜单结构，为该标签选项添加 dropdown 类，为下拉菜单结构添加 dropdown-menu 类便可以实现。

【例 20.6】（实例文件：ch20\Chap20.6.html）绑定导航和下拉菜单。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
       <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
       <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
       <script src="jquery.js"></script>
       <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
       <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <div class="container">
       <p>绑定导航和下拉菜单</p>
       <!--在ul中添加.nav和.nav-tabs，.nav-tabs用来设置标签页导航-->
       <ul class="nav nav-tabs">
          <!--在li中添加.nav-item，在a中添加.nav-link，对于选中的选项添加.active类
-->
```

```
            <!--添加data-toggle="tab"属性类别，是去掉a标签的默认行为，实现动态切换导航的active属性效果-->
            <li class="nav-item"><a class="nav-link" href="#">新闻</a></li>
            <!--.dropdown-toggle类和data-toggle="dropdown" 属性类别 用来激活下拉菜单-->
            <li class="nav-item"><a class="nav-link active dropdown-toggle" data-toggle="dropdown" href="#">教育</a>
                <!--.dropdown-menu用来指定被激活的菜单-->
                <ul class="dropdown-menu">
                    <li><a href="#" class="dropdown-item">初中</a></li>
                    <li><a href="#" class="dropdown-item">高中</a></li>
                    <li><a href="#" class="dropdown-item">大学</a></li>
                </ul>
            </li>
            <li class="nav-item"><a class="nav-link" href="#">旅游</a></li>
            <li class="nav-item"><a class="nav-link" href="#">美食</a></li>
            <li class="nav-item"><a class="nav-link" href="#">理财</a></li>
            <li class="nav-item"><a class="nav-link" href="#">招聘</a></li>
        </ul>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-13 所示。

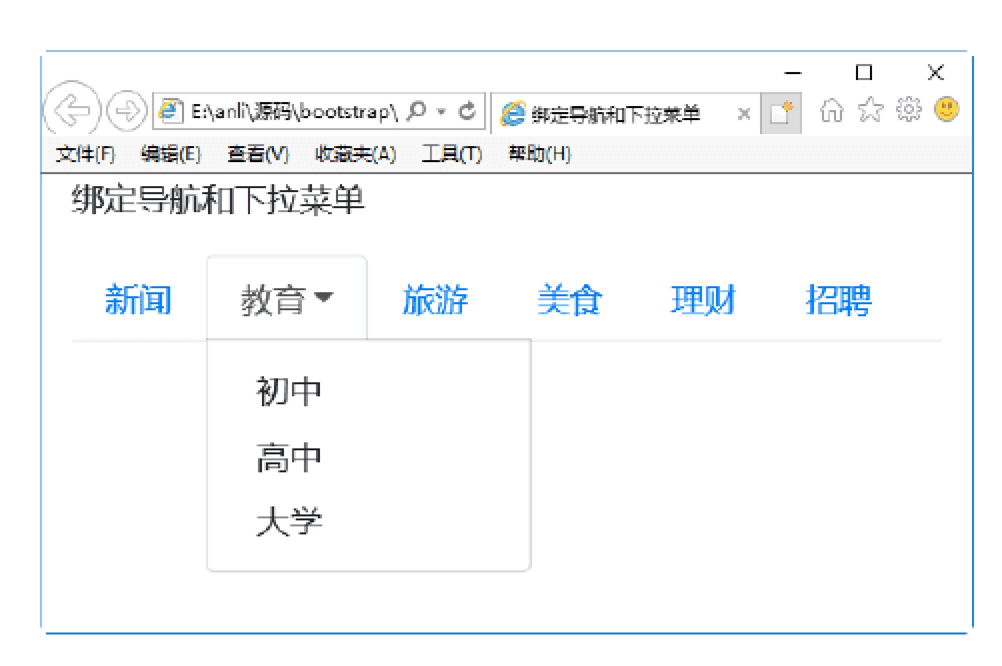


图 20-13　导航和下拉菜单绑定

20.4.5　面包屑导航

面包屑导航（Breadcrumbs）是一种基于网站层次信息的显示方式，它表示当前页面在导航层次结构内的位置。在 CSS 中利用::before 和 content 来添加分隔符。

【例 20.7】（实例文件：ch20\Chap20.7.html）面包屑。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title> </title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
```

```
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    <style>
            /*利用::before 和content添加分隔线*/
            li::before {
                padding-right: 0.5rem;
                padding-left: 0.5rem;
                color: #6c757d;
                content: ">";          /*添加分割线为“>” */
            }
            /*去掉第一个li前面的分隔线*/
            li:first-child::before {
                content: "";           /*设置第一个li元素前面为空*/
            }
    </style>
    </head>
    <body>
    <div class="container">
        <!--在ul中添加.breadcrumb类，设置面包屑-->
        <ul class="breadcrumb">
            <li><a href="#">学校</a></li>
            <li><a href="#">图书馆</a></li>
        </ul>
        <ul class="breadcrumb">
            <li><a href="#">学校</a></li>
            <li><a href="#">图书馆</a></li>
            <li><a href="#">图书</a></li>
        </ul>
        <ul class="breadcrumb">
            <li><a href="#">学校</a></li>
            <li><a href="#">图书馆</a></li>
            <li><a href="#">图书</a></li>
            <li><a href="#">编程类</a></li>
        </ul>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-14 所示。



图 20-14 面包屑导航

20.4.6 使用广告屏

通过在<div>元素中添加.jumbotron 类来创建 jumbotron（超大屏幕），它是一个大的灰色背景框，里面可以设置一些特殊的内容和信息，如 HTML 标签、Bootstrap 的元素等。如果想创建一个没有圆角的 jumbotron，就可以在.jumbotron-fluid 类的 div 中添加.container 或.container-fluid 类来实现。

【例 20.8】（实例文件：ch20\Chap20.8.html）广告屏。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title> </title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <!--添加.jumbotron类创建广告屏-->
    <div class="jumbotron">
        <h1>北京欢迎你!</h1>
        <p>北京，简称“京”，是中华人民共和国的首都，文化中心、科技创新中心。</p>
        <hr>
        <p>Beijing, or "jing" for short, It is the capital of the People's Republic
of China, cultural center、Technology innovation center.</p>
        <p>
            <!--.btn类为按钮添加基本样式，.btn-primary表示原始按钮样式（未被操作）-->
            <button class="btn btn-primary">了解更多</button>
        </p>
    </div>
```

```
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-15 所示。



图 20-15　广告屏组件

20.4.7　使用 card（卡片）

通过 Bootstrap 4 的.card 与.card-body 类来创建一个简单的卡片，如下面的代码所示。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <div class="container">
    <div class="card">
    <div class="card-body">简单的卡片</div>
    </div>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-16 所示。

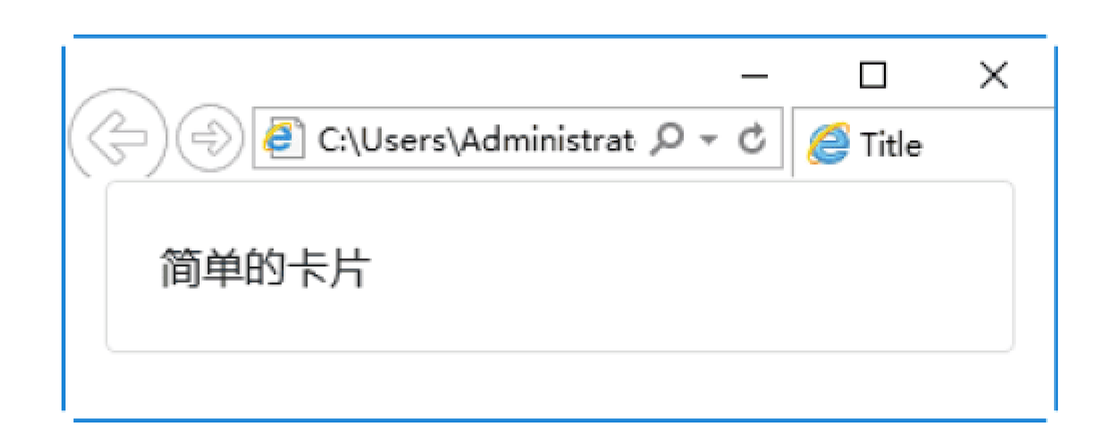


图 20-16　简单的卡片

卡片是一个灵活的、可扩展的内容窗口，其包含了可选的卡片头和卡片脚、一个大范围的内容、上下文背景色及强大的显示选项。卡片代替了 Bootstrap 3 中的 panel、well 和 thumbnail 等组件。

【例 20.9】（实例文件：ch20\Chap20.9.html）卡片。

```
<!DOCTYPE html>
<html>
<head>
<title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
</head>
<body>
<div class="container">
    <!--添加.card类创建卡片，.bg-success类设置卡片的背景颜色，.text-white设置卡片的文本颜色-->
    <div class="card bg-success text-white">
        <!--.card-header类用于创建卡片的头部样式-->
        <div class="card-header">卡片头</div>
        <div class="card-body">
            <!--给 <img> 添加 .card-img-top可以设置图片在文字上方或添加.card-img-bottom设置图片在文字下方。-->
            <img src="004.jpg" alt="" width="100%" height="200px">
            <h4 class="card-title">乡间小路</h4>
            <p class="card-text">太阳西下，黄昏下的乡村小路，弯弯曲曲延伸到村子的尽头，高低起伏的路面变幻莫测，只有叽叽喳喳在田间嬉闹的麻雀，此时也飞得无影无踪，大地只留下一片清凉。</p>
        </div>
        <!--.card-footer 类用于创建卡片的底部样式-->
        <div class="card-footer">卡片脚</div>
    </div>
</div>
</body>
</html>
```

在 IE 11.0 浏览器中运行的结果如图 20-17 所示。



图 20-17　卡片组件

20.4.8　使用进度条

进度条主要用来表示用户的任务进度，如下载、删除、复制等。

创建一个基本的进度条有以下 3 个步骤：

01 添加一个含有.progress 类的<div>。

02 在上面的<div>中添加一个含有.progress-bar 的空的<div>。

03 为含有.progress-bar 类的<div>添加一个带有百分比表示宽度的 style 属性，如 style="50%"，表示进度条在 50%的位置。

【例 20.10】（实例文件：ch20\Chap20.10.html）简单的进度条。

```
<!DOCTYPE html>
<html>
<head>
<title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
</head>
<body>
<div class="container">
    <p>基本的进度条</p>
    <div class="progress">
```

```
            <div class="progress-bar " style="width:50%"></div>
        </div>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-18 所示。

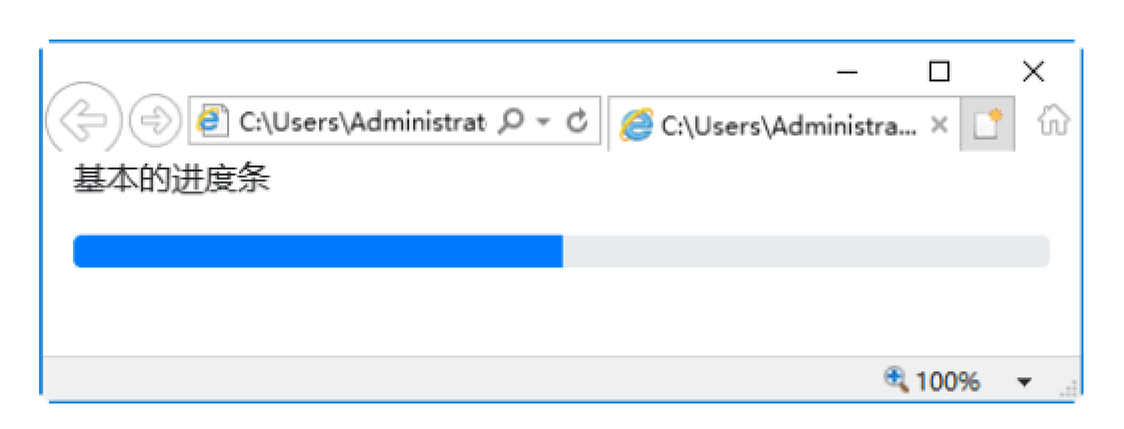


图 20-18　基本的进度条

1. 设置高度和添加文本

我们可以在基本进度条的基础上设置高度和添加文本，即在含有.progress 类的<div>中设置高度，在含有.progress-bar 类的<div>中添加文本内容。

【例 20.11】（实例文件：ch20\Chap20.11.html）设置高度和文本的进度条。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <div class="container">
        <p>设置高度和文本的进度条</p>
        <!--设置进度条高度20px，文本内容为--60%-->
        <div class="progress" style="height:20px">
            <div class="progress-bar " style="width:60%">60%</div>
        </div><br>
        <!--设置进度条高度为30px，文本内容为--80%-->
        <div class="progress" style="height:30px">
            <div class="progress-bar " style="width:80%">80%</div>
        </div>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-19 所示。

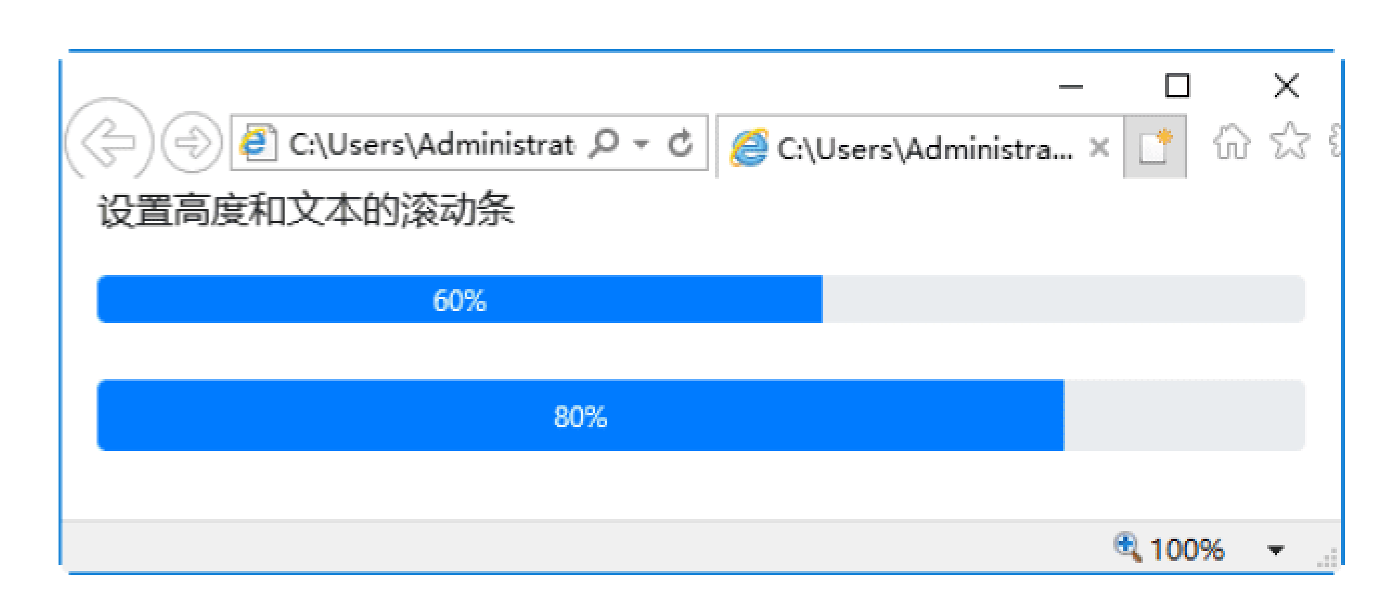


图 20-19　设置高度和添加文本

2. 设置不同的背景颜色

进度条的默认背景颜色是蓝色，为了能给用户一个更好的体验，进度条和警告信息框一样，也可以根据不同的状态配置不同的进度条颜色。我们可以通过添加 bg-success、bg-info、bg-warning 和 bg-danger 类来改变默认背景颜色，分别表示浅绿色、浅蓝色、浅黄色和浅红色。

【例 20.12】（实例文件：ch20\Chap20.12.html）设置不同的背景颜色。

```
<!DOCTYPE html>
<html>
<head>
<title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
    <script src="jquery.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
</head>
<body>
<div class="container">
    <p>不同颜色的进度条</p>
    <div class="progress">
        <div class="progress-bar" style="width:30%">默认</div>
    </div>
    <br>
    <div class="progress">
        <div class="progress-bar bg-success"
style="width:40%">bg-success</div>
    </div>
    <br>
    <div class="progress">
        <div class="progress-bar bg-info" style="width:50%">bg-info</div>
    </div>
    <br>
```

```
        <div class="progress">
            <div class="progress-bar bg-warning"
style="width:60%">bg-warning</div>
        </div>
        <br>
        <div class="progress">
            <div class="progress-bar bg-danger"
style="width:70%">bg-danger</div>
        </div>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-20 所示。

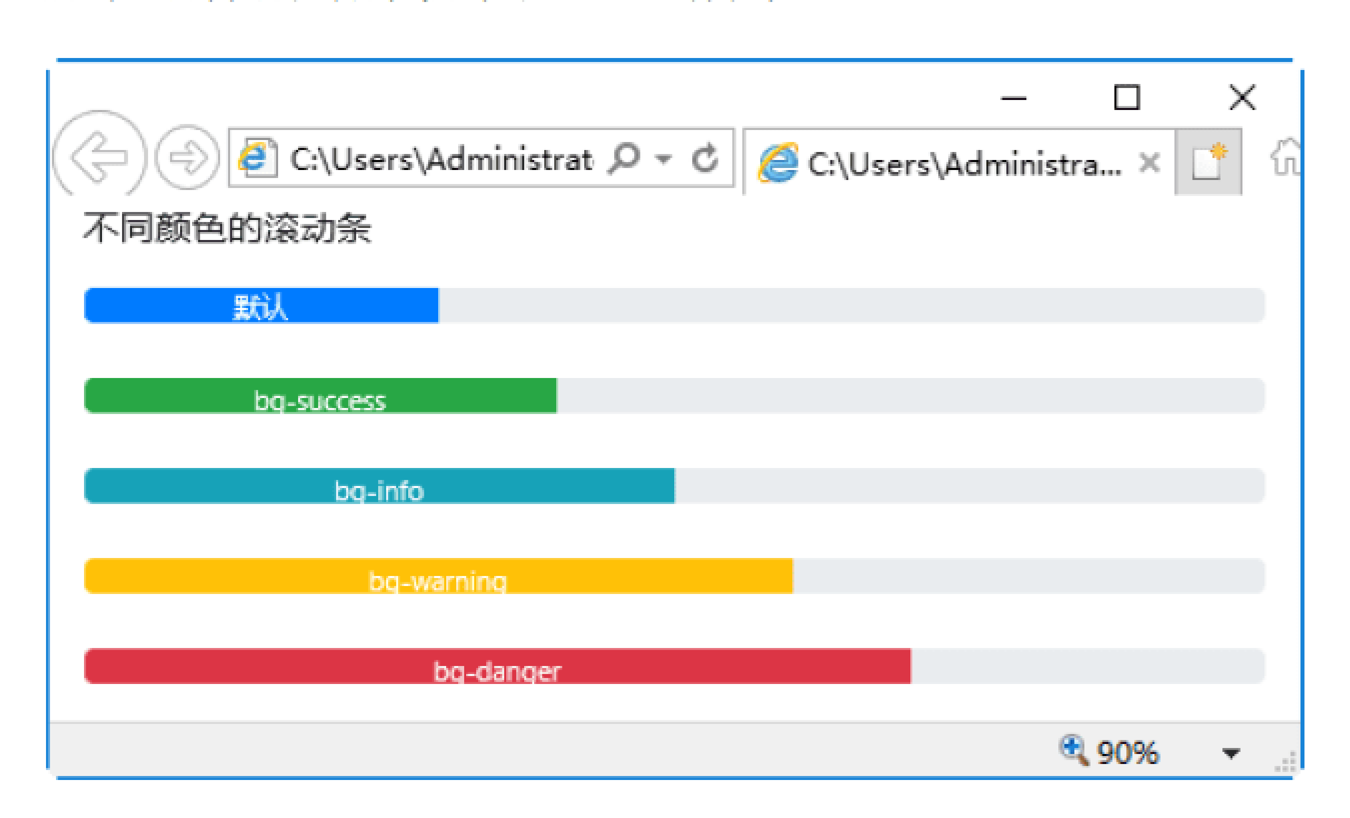


图 20-20　不同背景的进度条

3. 设置动画条纹进度条

还可以为进度条添加 progress-bar-striped 和 progress-bar-animated 类，即为滚动条添加彩色条纹和动画效果。

【例 20.13】（实例文件：ch20\Chap20.13.html）设置动画条纹进度条。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <div class="container">
```

```
        <p>设置进度条纹效果</p>
        <!--添加.progress类，创建进度条-->
        <div class="progress">
            <!--.progress-bar-striped类设置进度条条纹效果，.progress-bar-animated
类设置条纹进度条的动画效果-->
            <div class="progress-bar progress-bar-striped
progress-bar-animated" style="width:60%"></div>
        </div>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-21 所示。

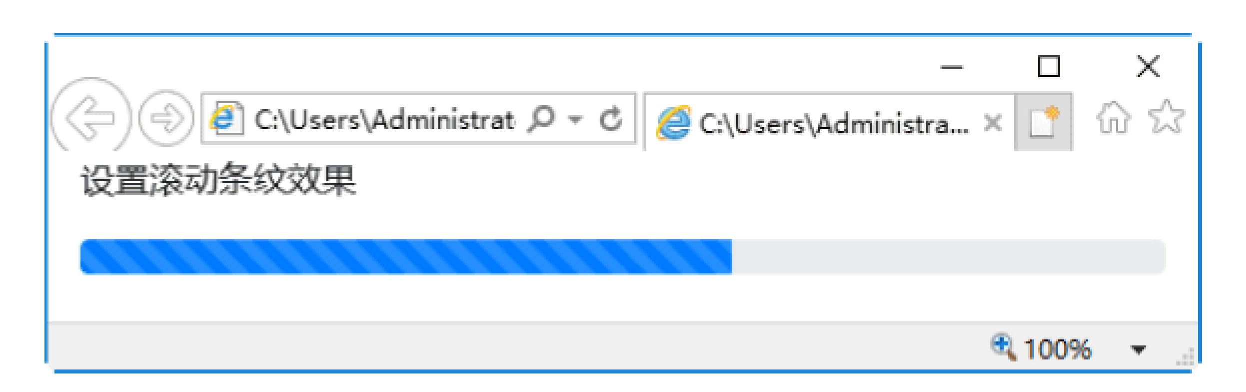


图 20-21　带条纹的进度条

4. 混合色彩的进度条

在进度条中，我们可以在含有.progress 类的<div>中添加多个含有.progress-bar 类的<div>，然后分别为每个含有.progress-bar 类的<div>设置不同的背景颜色来实现混合色彩的进度条。

【例 20.14】（实例文件：ch20\Chap20.14.html）混合色彩的进度条。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <div class="container">
        <p>混合色彩的进度条</p>
        <div class="progress" style="height:30px">
            <div class="progress-bar bg-success"
style="width:20%">bg-success</div>
            <div class="progress-bar bg-info" style="width:20%">bg-info</div>
            <div class="progress-bar bg-warning"
style="width:20%">bg-warning</div>
```

```
            <div class="progress-bar bg-danger"
style="width:20%">bg-danger</div>
        </div>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-22 所示。

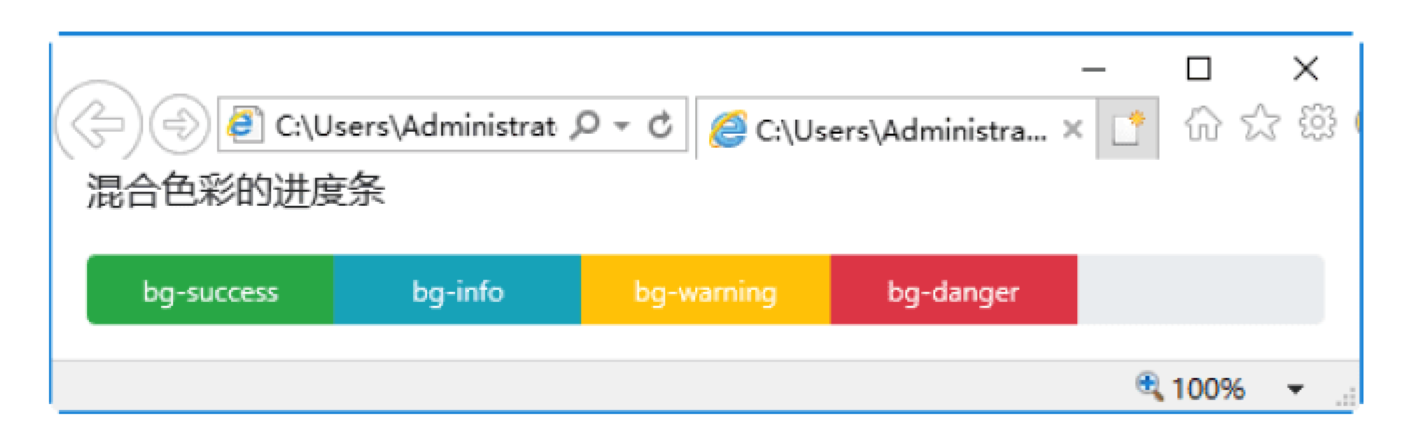


图 20-22　混合色彩的进度条

20.4.9　使用模态框

模块框是一种灵活的、对话框式的提示，是页面的一部分，是覆盖在父窗体上的子窗体。通常是显示来自一个单独源的内容，可以在不离开父窗体的情况下有一些互动。

模态框的基本结构如下面的代码所示。

```
    <!--按钮——用于打开模态框-->
    <button type="button" data-toggle="modal"
data-target="#myModal">...</button>
    <!--定义模态框-->
    <div class="modal fade" id="myModal">
           <div class="modal-dialog">
              <div class="modal-content">
                 <div class="modal-header">...</div>
                 <div class="modal-body">...</div>
                 <div class="modal-footer">...</div>
              </div>
           </div>
        </div>
    </div>
```

在上面的结构中，按钮中的属性类别分析如下。

（1）data-toggle="modal"：用于打开模态框。

（2）data-target="#myModal"：指定打开的模态框目标（使用哪个模态框就把哪个模态框的 id 写在其中）。

定义模态框中的属性类别分析如下。

（1）.modal 类：用来把<div>的内容识别为模态框。

（2）.fade 类：当模态框被切换时，设置模态框的淡入淡出。

（3）id="myModal"：被指定打开的目标 id。

（4）.modal-dialog：定义模态对话框层。

（5）.modal-content：定义模态对话框的样式。

（6）.modal-header：为模态框的头部定义样式的类。

（7）.modal-body：为模态框的主体定义样式的类。

（8）.modal-footer：为模态框的底部定义样式的类。

（9）data-dismiss="modal"：用于关闭模态窗口。

【例 20.15】（实例文件：ch20\Chap20.15.html）模态框。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <div class="container">
    <h3>模态框</h3>
    <!-- 按钮：用于打开模态框 -->
    <button type="button" class="btn btn-primary" data-toggle="modal"
data-target="#myModal">
          打开模态框
     </button>
    <!-- 模态框 -->
    <div class="modal fade" id="myModal">
        <div class="modal-dialog">
            <div class="modal-content">
                <!-- 模态框头部 -->
                <div class="modal-header">
                    <!--modal-title用于设置标题在模态框头部垂直居中-->
                    <h4 class="modal-title">用户注册</h4>
                    <button type="button" class="close"
data-dismiss="modal">&times;</button>
                </div>
                <!-- 模态框主体 -->
                <div class="modal-body">
                    <form action="#">
                        <p>姓名：<input type="text"></p>
                        <p>密码：<input type="password"></p>
```

```
                    <p>邮箱：<input type="email"></p>
                </form>
            </div>
            <!-- 模态框底部 -->
            <div class="modal-footer">
                <button type="button" class="btn btn-primary">提交</button>
                <button type="button" class="btn btn-secondary" 
data-dismiss="modal">
                    关闭
                </button>
            </div>
        </div>
    </div>
</div>
</div>
</body>
</html>
```

在 IE 11.0 浏览器中运行的结果如图 20-23 所示；单击“打开模态框”按钮激活模态框，效果如图 20-24 所示。

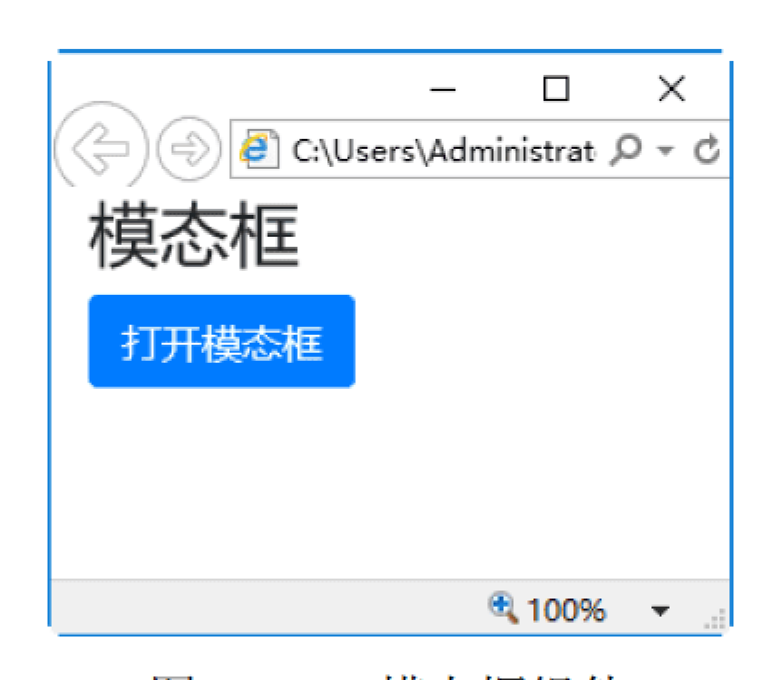


图 20-23　模态框组件



图 20-24　打开模态框效果

20.4.10　使用滚动监听

滚动监听即根据滚动条的位置自动更新对应的导航目标。

实现滚动监听可分为以下三步：

（1）设计导航栏及可滚动的元素，可滚动元素上的 id 值要匹配导航栏上超链接的 href 属性，若可滚动元素的 id 属性值为“a”，则导航栏上超链接的 href 属性值应该为“#a”。

（2）为想要监听的元素添加 data-spy="scroll"属性类别，然后添加 data-target 属性，它的值为导航栏的 id 或 class 值，这样才可以联系上可滚动的区域。监听的元素通常是<body>。

（3）设置相对定位，使用 data-spy="scroll"的元素需要将其 CSS 的 position 属性设置为 relative 才能起作用。

data-offset 属性用于计算滚动位置时距离顶部的偏移像素，默认为 10 像素。

【例 20.16】（实例文件：ch20\Chap20.16.html）滚动监听。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
        <meta name="viewport" content="width=device-width, initial-scale=1, 
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
    <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    <style>
    body {
        position: relative;
    }
    #navbar{
          position: fixed;
          top:200px;
           right: 50px;
    }
    </style>
    </head>
    <!--添加data-spy="scroll" 属性类别，设置监听元素-->
    <!--data-target="#navbar"属性类别指定导航栏的id（navbar）-->
    <body data-spy="scroll" data-target="#navbar" data-offset="50">
    <!--.navbar设置导航，.bg-dark类和.nav-dark类设置黑色背景、白色文字-->
    <nav class="navbar bg-dark navbar-dark" id="navbar">
        <!--.navbar-nav是在导航.nav的基础上重新调整了菜单项的浮动与内外边距。-->
        <ul class="navbar-nav">
            <!--在li中添加.nav-item ,在a中添加.nav-link设置导航的样式-->
            <li class="nav-item">
                <a class="nav-link" href="#s1">Section 1</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#s2">Section 2</a>
            </li>
            <li class="nav-item">
                <!--.dropdown-toggle类和data-toggle="dropdown" 属性类别，用来激活
下拉菜单-->
                <a class="nav-link dropdown-toggle" data-toggle="dropdown" 
href="#">
                    Section 3
                </a>
                <!--.dropdown-menu用来指定被激活的菜单-->
                <div class="dropdown-menu">
                    <!--.dropdown-item添加列表元素的样式-->
                    <a class="dropdown-item" href="#s3">3.1</a>
                    <a class="dropdown-item" href="#s4">3.2</a>
                </div>
            </li>
```

```
        </ul>
    </nav>
    <div id="s1">
        <h1>Section 1</h1>
        <p><img src="005.jpg" alt="" width="300px" height="300px"></p>
    </div>
    <div id="s2">
        <h1>Section 2</h1>
        <p><img src="006.jpg" alt="" width="300px" height="300px"></p>
    </div>
    <div id="s3">
        <h1>Section 3.1</h1>
        <p><img src="007.jpg" alt="" width="300px" height="300px"></p>
    </div>
    <div id="s4">
        <h1>Section 3.2</h1>
        <p><img src="008.jpg" alt="" width="300px" height="300px"></p>
    </div>
    </body>
    </html>
```

在 IE 11.0 浏览器中运行的结果如图 20-25 所示。当滚动滚动条时，导航条会实时监听并更新当前被激活的菜单项，效果如图 20-26 所示。

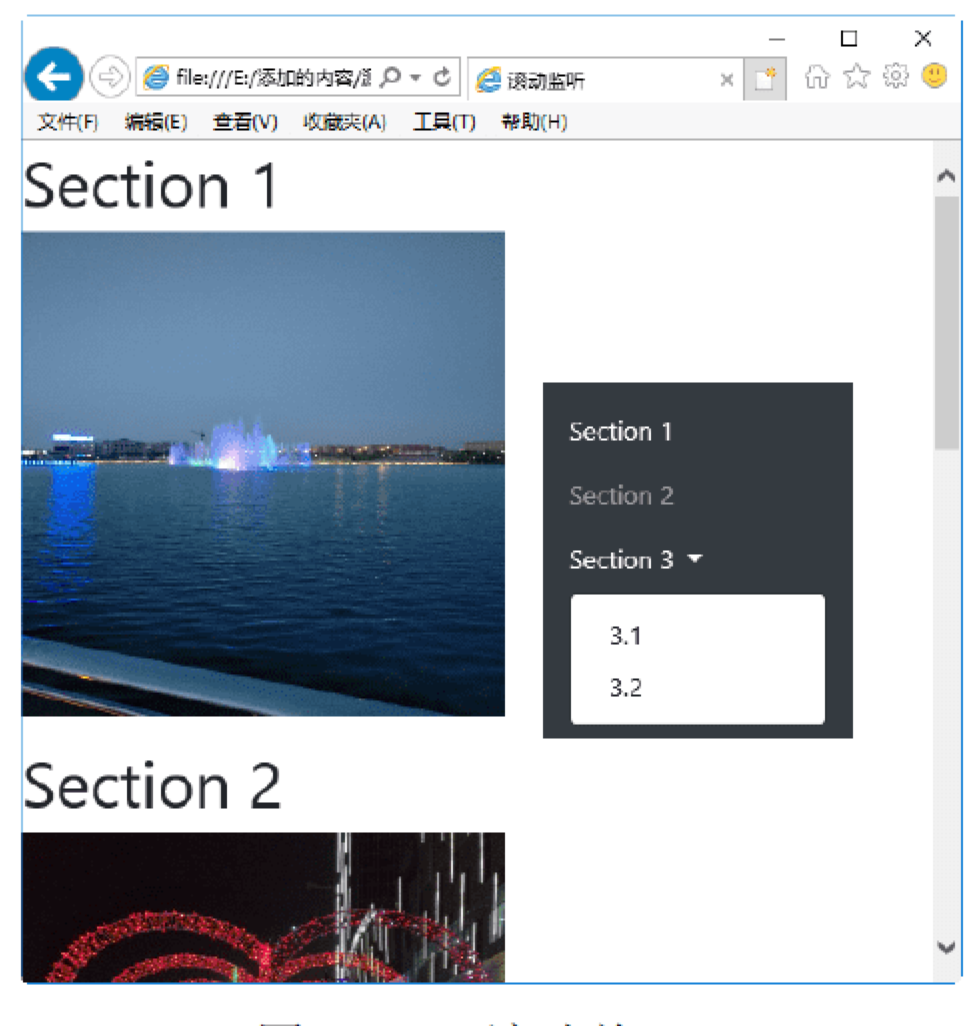


图 20-25　滚动前

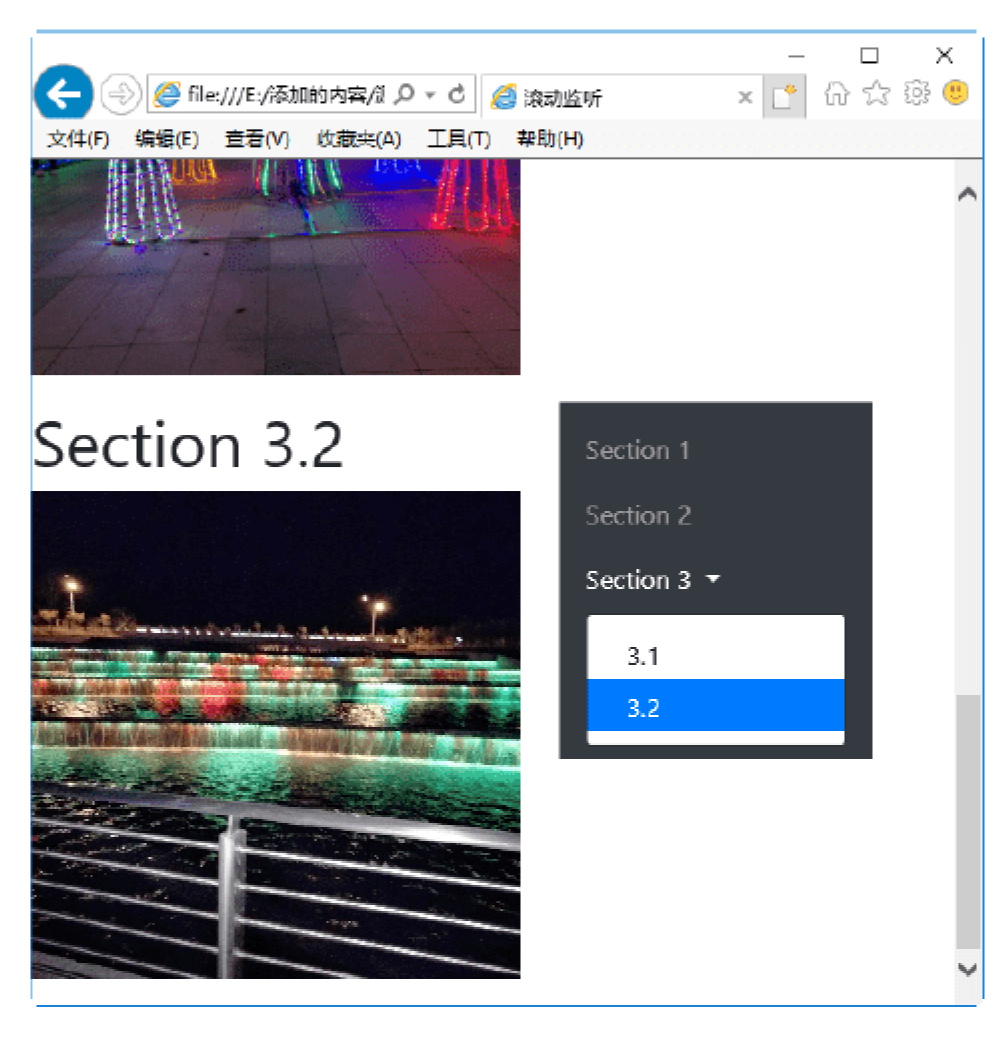


图 20-26　滚动后

20.5　案例实战——胶囊导航选项卡（Tab 栏）

选项卡是网页中一种常用的功能，当用户点击或悬浮对应的菜单选项时，能切换出对应的内容。

使用 Bootstrap 框架实现胶囊导航选项卡只需要以下两部分内容便可完成。

（1）胶囊导航组件：对应的是 Bootstrap 中的 nav-pills。

（2）可以切换的选项卡面板：对应的是 Bootstrap 中的 tab-pane 类。选项卡面板的内容统一放在 tab-content 容器中，而且每个内容面板 tab-pane 都需要设置一个独立的选择符（ID）与选项卡中 data-target 或 href 的值匹配。

注意，选项卡中链接的锚点要与对应的面板内容容器的 ID 相匹配。

【例 20.17】（实例文件：ch20\Chap20.17.html）胶囊导航选项卡。

```
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <script src="jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
    <body>
    <div class="container">
        <h2>胶囊导航选项卡</h2>
        <!--在ul中添加.nav和.nav-pills，.nav-pills类用来设置胶囊导航-->
        <ul class="nav nav-pills">
            <!--在li中添加.nav-item，在a中添加.nav-link，对于选中的选项添加.active类
-->
            <!--添加data-toggle="pill"属性类别，是去掉a标签的默认行为，实现动态切换导
航的active属性效果-->
            <!--给每个a标签的href属性添加属性值，用于绑定下面选项卡面板中对应的元素，当导
航切换时，显示对应的内容-->
            <li class="nav-item"><a class="nav-link active" data-toggle="pill"
href="#tab1">图片1</a></li>
            <li class="nav-item"><a class="nav-link" data-toggle="pill"
href="#tab2">图片2</a></li>
            <li class="nav-item"><a class="nav-link" data-toggle="pill"
href="#tab3">图片3</a></li>
            <li class="nav-item"><a class="nav-link" data-toggle="pill"
href="#tab4">图片4</a></li>
        </ul>
        <!--选项卡面板-->
        <!-- 选项卡面板中tab-content类和.tab-pane类 与data-toggle="pill"一同使用，
设置标签页对应的内容随胶囊导航的切换而更改-->
        <div class="tab-content">
            <!--.active类用来设置胶囊导航默认情况下激活的选项所对应的元素-->
            <div id="tab1" class="tab-pane active">
                <img src="01.png" alt="景色1" class="img-fluid">
            </div>
            <div id="tab2" class="tab-pane fade">
                <img src="02.png" alt="景色2" class="img-fluid">
```

```
        </div>
        <div id="tab3" class="tab-pane fade">
            <img src="03.png" alt="景色3" class="img-fluid">
        </div>
        <div id="tab4" class="tab-pane fade">
            <img src="04.png" alt="景色4" class="img-fluid">
        </div>
    </div>
</div>
</body>
</html>
```

在 IE 11.0 浏览器中运行的结果如图 20-27 所示。单击“nav4”选项卡切换面板内容，效果如图 20-28 所示。



图 20-27　页面加载完成后效果



图 20-28　单击“nav4”选择卡

20.6　专家解惑

1. 如何使用 Bootstrap 创建缩略图？

使用 Bootstrap 创建缩略图的步骤如下：

（1）在图像周围添加带有 class .thumbnail 的 <a> 标签。

（2）添加 4 个像素的内边距（padding）和一个灰色的边框。

（3）当鼠标悬停在图像上时，会动画显示出图像的轮廓。

2. 如何使用 Bootstrap 实现轮播效果？

Bootstrap 轮播（Carousel）插件是一种灵活的响应式的向站点添加滑块的方式。除此之外，内容也是足够灵活的，可以是图像、内嵌框架、视频或其他想要放置的任何类型的内容。

例如以下代码是实现一个简单的图片轮播效果。

```
    <div id="myCarousel" class="carousel slide">
        <!-- 轮播（Carousel）指标 -->
        <ol class="carousel-indicators">
            <li data-target="#myCarousel" data-slide-to="0"
class="active"></li>
            <li data-target="#myCarousel" data-slide-to="1"></li>
            <li data-target="#myCarousel" data-slide-to="2"></li>
        </ol>
        <!-- 轮播（Carousel）项目 -->
        <div class="carousel-inner">
            <div class="item active">
                <img src="01.png" alt="第1幅图">
            </div>
            <div class="item">
                <img src="02.png" alt="第2幅图">
            </div>
            <div class="item">
                <img src="03.png" alt="第3幅图">
            </div>
        </div>
        <!-- 轮播（Carousel）导航 -->
        <a class="left carousel-control" href="#myCarousel" role="button"
data-slide="prev">
            <span class="glyphicon glyphicon-chevron-left"
aria-hidden="true"></span>
            <span class="sr-only">Previous</span>
        </a>
        <a class="right carousel-control" href="#myCarousel" role="button"
data-slide="next">
            <span class="glyphicon glyphicon-chevron-right"
aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
    </div>
```

效果如图 20-29 所示。

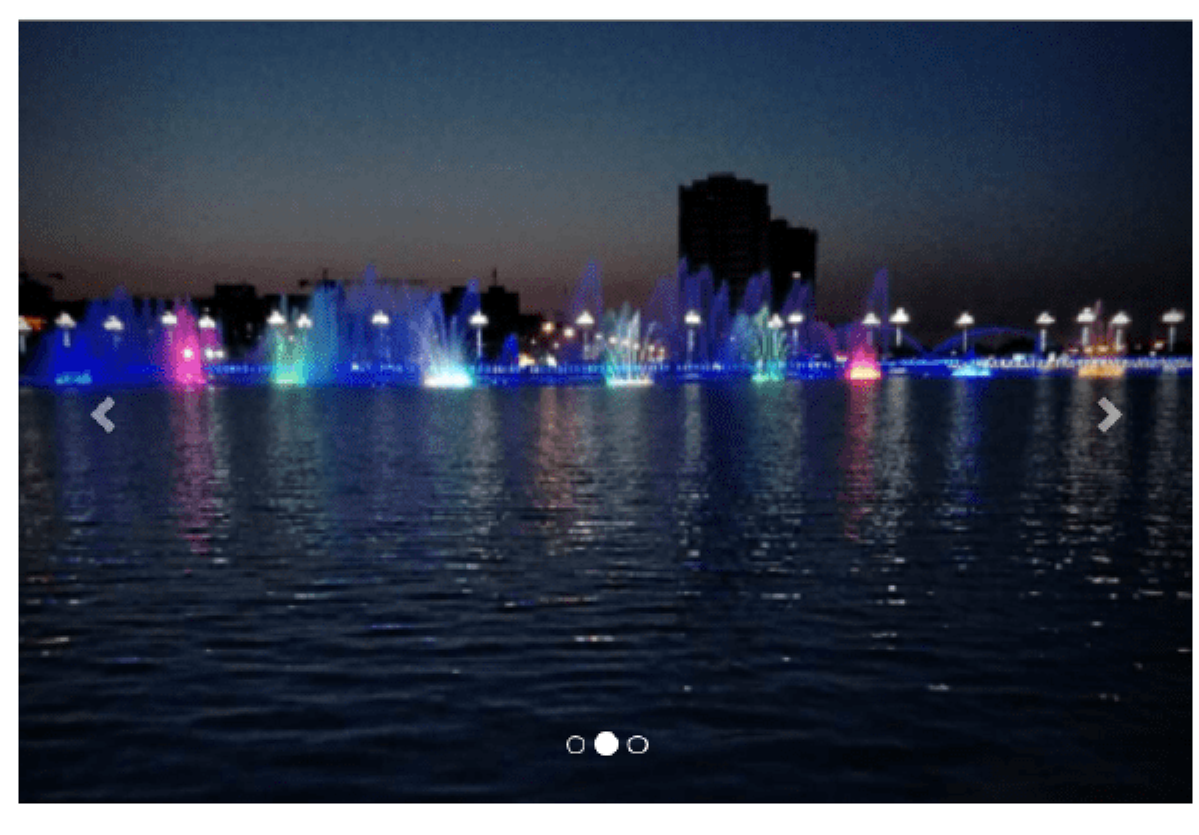


图 20-29　图片轮播效果

第21章　开发计算器APP

本章将介绍一个简洁的计算器项目，该项目涉及HTML5移动应用程序的开发操作和流程，最终完成一个APP的开发，读者可了解其设计思路和交互实现。

21.1　项目概述

21.1.1　功能梳理

该项目介绍一款简单易用的计算机应用程序，它是使用纯前端技术开发的，是一个HTML5版本的移动APP，可通过HBuilder IDE打包成可在手机上安装的APK安装包。程序在手机上安装后，打开操作界面直接进入应用主页面，用户便可以使用该APP提供的功能了。该APP提供的功能如下：

（1）基本的计算器运算功能：加、减、乘、除、取余、清空、后退等。

（2）历史运算记录查看的功能。

（3）“关于”页面的展示。

（4）按键区集成拨号打电话的功能。

21.1.2　开发环境

该案例的开发环境配置如下。

（1）系统：Windows 10。

（2）IDE：　Hbuilder。

（3）开发语言：HTML5、JavaScript、CSS3。

开发环境的搭建比较简单，在HBuilder官网下载对应的IDE，运行项目即可。

21.1.3 项目结构目录

使用 HBuilder 打开项目之后，整个项目的代码结构如图 21-1 所示。

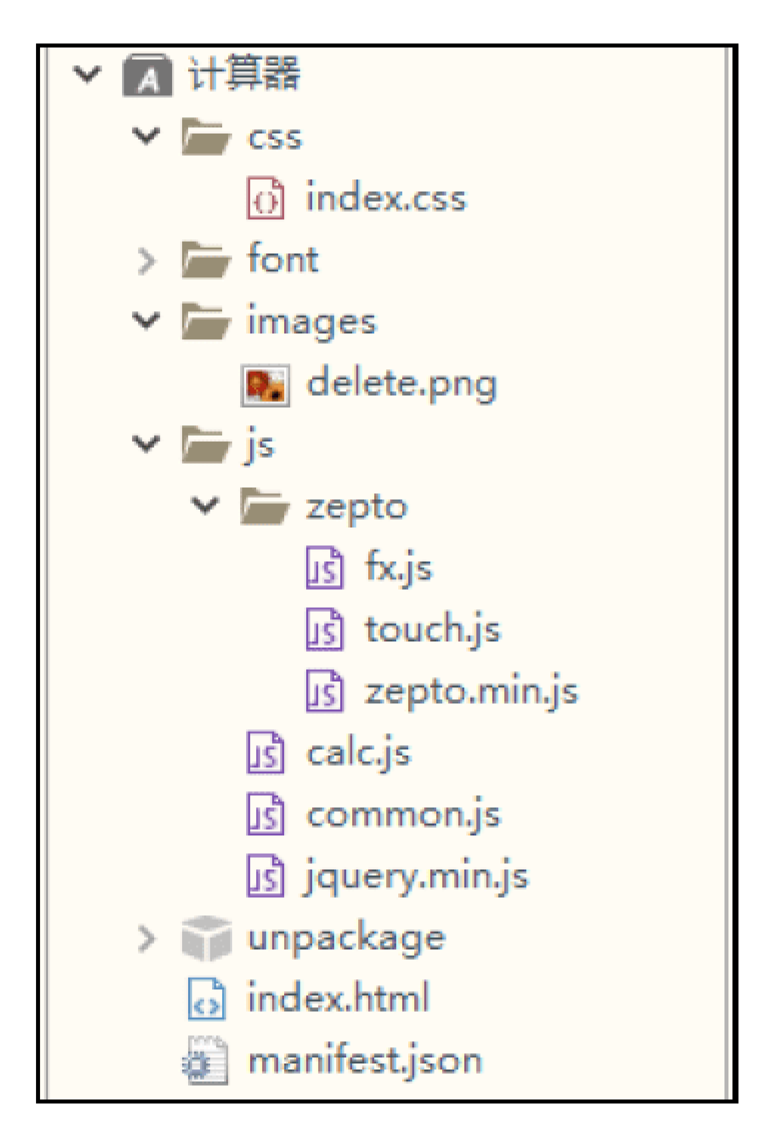


图 21-1 计算器项目目录

该项目的代码目录按照 HBuilder 项目文件结构组织。按照不同任务的功能分配主要包括：

（1）css 包目录下包含 1 个文件：index.css 是本项目的样式文件。

（2）font 包文件：该目录下的代码文件是本项目中所用到的字体和字体图标文件。

（3）image 包目录下包含 1 个文件：delete.png 是本项目中所用的图片资源。

（4） js 包目录下包含 4 个文件：zepto 包目录包含了该 APP 所须的其他 JavaScript 框架，calc.js 是该 APP 中相应的运算或数据交互的处理，common.js 是所须的工具类，jquery.min.js 是用到的 jQuery 框架。

（5） unpackage 文件夹：指定不被打包的目录或文件。

（6） index.html：该项目的主页。

（7） manifest.json 文件：本项目的配置文件，内容涉及所需权限等。

21.1.4 项目打包

HBuilder 默认是在云端打包的，也就是将代码提交上去进行打包，然后下载打好的包。项目打包优点：不管机器配置高低，只要网速快都可以很快打包好，当然也可以进行本地打包，那样就需要 Android 环境和 IOS 环境，这里不做推荐。

在 HBuilder 编辑器中选择“发行”|“原生 APP-云端打包”命令，如图 21-2 所示。



图 21-2　HBuilder 编辑器

进入到“APP 云端打包”窗口，如图 21-3 所示。

图 21-3　APP 云端打包

具体的选择步骤如下：

1. 选择平台

选择 Android 或 IOS，或者两者都有。

2. 选择证书

如果只是练习，可以选择公用证书，但是这样不能发到线上；如果要发布到线上，就需要自己申请 google 和 Apple 的证书。

3. 确认打包

单击“打包”按钮，系统会显示打包进度。当打包完成后可以手动选择下载到本地地址，下载完成后就可以安装到手机运行了。

21.1.5　项目效果展示

打包完成后，把打包好的 APK 导入到手机并安装，便可以使用了。下面来展示一下项目的效果。

首先在手机中打开软件，页面效果如图 21-4 所示。

界面的右上方有两个小按钮，当点击左边小按钮时，将弹出关于计算器的说明，效果如图 21-5 所示；当点击右边小按钮时，将弹出历史记录，效果如图 21-6 所示；当点击右下角的“删除”按钮时，可以删除历史记录，效果如图 21-7 所示。



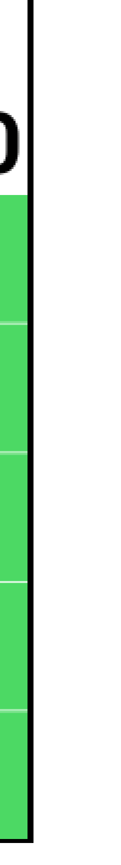


图 21-4　计算器主页

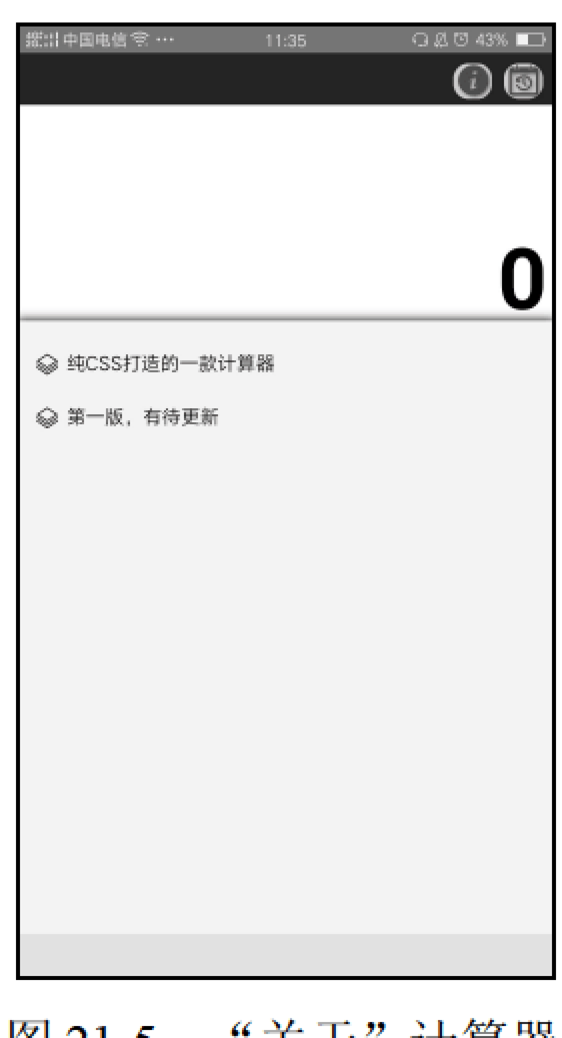


图 21-5　“关于”计算器

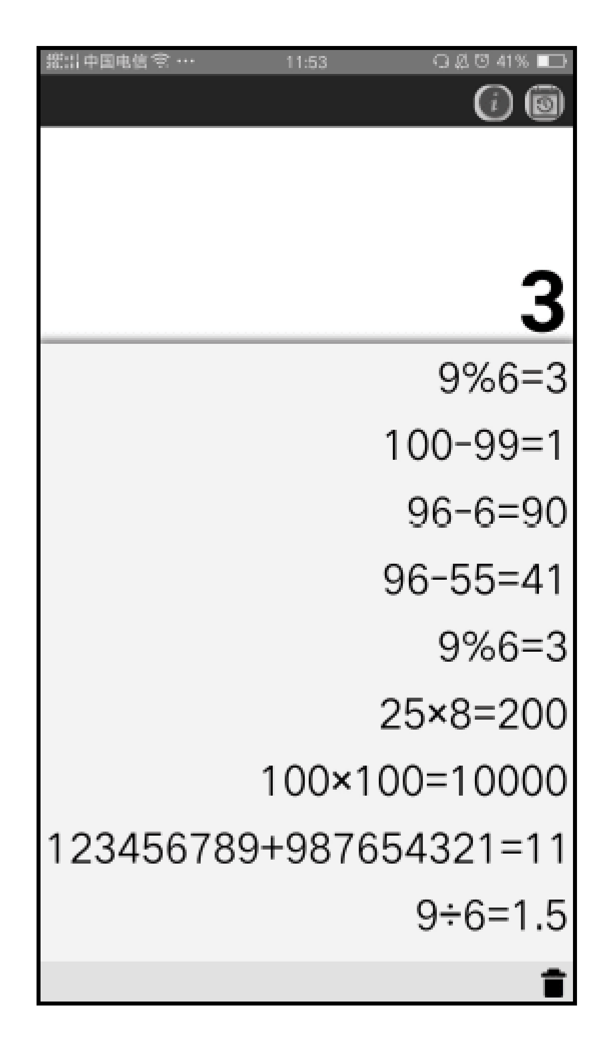


图 21-6　历史记录

该计算器还有一个功能，当输入手机号码后长按'='可以拨打电话，如图 21-8 所示。

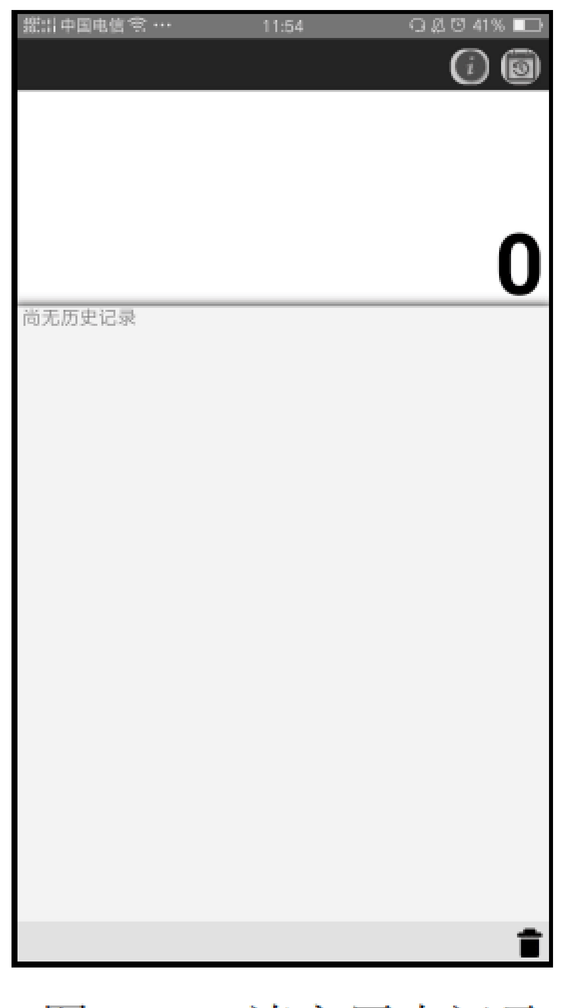


图 21-7　清空历史记录



图 21-8　输入手机号

21.2　项目设计

本项目主要由 index.html、index.css、calc.js 和 common.js 等文件组成，具体内容请参照下面的代码。其他工具类的代码不再一一赘述。

21.2.1 index.html 文件

index.html 文件是项目的主页面，具体代码如下所示。

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <link rel="stylesheet" href="css/index.css" />
    <script src="js/zepto/zepto.min.js"></script>
    <script src="js/zepto/fx.js"></script>
    <script src="js/zepto/touch.js"></script>
    <script src="js/common.js"></script>
    <script src="js/calc.js"></script>
</head>
<body>
<div id="container">
    <div id="calc" class="calc">
        <div id="top">
            <div id="win-tool">
                <a id="about" href="javascript:;" title="关于"><i id="au" class="iconfont about">&#xe610;</i></a>
                <a id="history" href="javascript:void(0);" title="历史"><i id="h" class="iconfont history">&#xe60d;</i></a>
            </div>
            <div id="result">
                <div id="express"></div>
                <div id="res">0</div>
            </div>
        </div>
        <div id="bottom">
            <div class="row">
                <span id="reset" data-number="C">CE</span>
                <span id="remove" data-number="←">←</span>
                <span data-number="%">%</span>
                <span data-number="/" class="tool chu">÷</span>
            </div>
            <div class="row">
                <span data-number="7">7</span>
                <span data-number="8">8</span>
                <span data-number="9">9</span>
                <span data-number="*" class="tool cheng">×</span>
            </div>
            <div class="row">
                <span data-number="4">4</span>
                <span data-number="5">5</span>
                <span data-number="6">6</span>
```

```
                    <span data-number="-" class="tool jiang">—</span>
                </div>
                <div class="row">
                    <span data-number="1">1</span>
                    <span data-number="2">2</span>
                    <span data-number="3">3</span>
                    <span data-number="+" class="tool add">+</span>
                </div>
                <div class="row">
                    <span data-number="0" class="zero">0</span>
                    <span data-number="." class="dian">.</span>
                    <span id="equals" data-number="=" class="tool eq">=
</span>
                </div>
            </div>
            <div id="historyBox">
                <div class="con">
                    <ul></ul>
                </div>
                <div class="remove">
                    <a href="javascript:;" title="清空历史记录"><i
class="iconfont del">&#xe613;</i></a>
                </div>
            </div>
        </div>
    </div>
    <!--移动端拨号功能-->
    <a href="#" id="telPhone" data-flag = ""></a>
    </body>
    </html>
```

21.2.2 index.css 文件

index.css 文件是项目的样式文件，具体代码如下所示。

```
    @font-face {
       font-family: 'iconfont';
       src: url('../font/iconfont.eot'); /* IE9*/
       src: url('../font/iconfont.eot?#iefix') format('embedded-opentype'),
/* IE6-IE8 */
       url('../font/iconfont.woff') format('woff'), /* chrome、firefox */
       url('../font/iconfont.svg#iconfont') format('svg'); /* iOS 4.1- */
       url('../font/iconfont.ttf') format('truetype'), /* chrome、firefox、
opera、Safari, Android, iOS 4.2+*/
    }
    .iconfont{
       font-family:"iconfont" !important;
       font-size:1.9rem;
```

```
    font-style:normal;
    -Webkit-font-smoothing: antialiased;
    -Webkit-text-stroke-width: 0.2px;
    -moz-osx-font-smoothing: grayscale;
    vertical-align: middle;
}
html{
    height: 100%;
    font-size: 62.5%;
}
.maxhtml {font-size: 82.5%;}
body{
     height: 100%;
    min-width: 300px;
    position: relative;
    padding: 0;
    margin: 0;
    font-size: 62.5%;
}
ul {padding: 0;margin: 0;}
#container {
    overflow: hidden;
    padding: 5px 5px;
}
.calc {
    width: 300px;
    max-width: 600px;
    margin: 0px auto;
    display: flex;
    flex-direction: column;
    border-radius: 5px;
    flex-wrap: wrap;
    height: 100%;
    transition: all .5s;
    overflow: hidden;
    z-index: 0;
    position: relative;
}
/*计算器顶部工具栏的样式*/
#top {
    display: flex;
    justify-content: flex-start;
    flex-direction: column;
    position: relative;
    width: 100%;
    height: 30%;
}
/*工具栏中按钮的样式*/
```

```
#win-tool {
    background-color: #242424;
    min-height: 35px;
    position: relative;
    padding: 7px 8px 0 8px;
    text-align: right;
    width: 100%;
    box-sizing: border-box;
}
#win-tool a {
    background: hsl(0, 0%, 80%);
    display: inline-block;
    width: 2rem;
    border-radius: 10px;
    text-align: center;
    color: inherit;
    text-decoration: none;
    font-family: "微软雅黑";
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.5); /*设置阴影*/
    margin-left: .5rem;
}
/*计算器数据显示的样式*/
#result {
    min-height: 90px;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: flex-end;
    color: #000;
    overflow: auto;
    box-sizing: border-box;
    text-align: right;
}
#result > #express {
    font-size: 2rem;
}
#result div {
     padding: 0 5px;
     box-sizing: border-box;
}
#result > #res {
    font-size: 4.5rem;
    font-weight: 600;
    width: 100%;
}
/*计算器键盘的样式设计*/
#bottom {
```

```
        font-weight: 600;
        width: 100%;
        height: 70%;
    }
    #bottom .row {
        height: 20%;
        width: 100%;
        display: flex;
        box-sizing: border-box;
        margin-bottom: -1px;
    }
    #bottom .row span:first-child {
        border-left: 0;
    }
    #bottom .row:last-child span {
        border-bottom: 0;
    }
    #bottom .row span{
        height: 100%;
        width: 25%;
        background-color: #AAAAAA;
        border: 1px solid #fff;
        border-right: 0;
        color:hsl(0, 17%, 20%);
         font-size: 2rem;
         line-height: 5rem;
        cursor: default;
        -Webkit-user-select: none;
        display: flex;
        align-items: center;
        justify-content: center;
    }
    /*设置加、减、乘、除、等号的样式*/
    #bottom .row span.tool {
        background-color: #4CD964;
        color: #fff;
    }
    /*清空历史记录页面样式*/
    #historyBox {
        background: hsl(0, 0%, 95%);
        width: 100%;
        height: 71%;
        position: absolute;
        z-index: 10;
        left: 0;
        bottom: 0;
        box-shadow: 0px 0px 10px #000;
        transition: all .3s;
```

```
        transform: translateY(102%);
        font-family:"微软雅黑";
    }
    #historyBox .con {
        height: 91%;
        overflow-y: auto;
    }
    #historyBox .remove {
         height: 28px;
        background: hsl(0, 0%, 88%);
        text-align: right;
        box-sizing: border-box;
        border-top: 1px solid transparent;
        position: absolute;
        width: 100%;
        left: 0;
        bottom: 0;
    }
    /*清空历史记录div标签中a的样式*/
    #historyBox .remove a{
         color: inherit;
        text-decoration: none;
        padding-right: 5px;
    }
    /*设置历史记录中ul的样式*/
    #historyBox ul {
        padding: 0 5px 0 5px;
        color: hsl(0, 0%, 50%);
        font-size: 1rem;
    }
    /*设置历史记录中li的样式*/
    #historyBox ul li {
        height: 45px;
        text-align: right;
        line-height: 45px;
        font-size: 2.2rem;
        font-family:"微软雅黑";
        background: hsl(0, 0%, 95%);
        color: #000;
        overflow: auto;
    }
    /*使用媒体查询来设计页面屏幕宽度小于500px的页面样式*/
    @media (max-width: 500px){
        html {
         font-size: 80%;
         font-family:Helvetica;
        }
        .iconfont {font-size: 1.6rem;}
```

```
        #container {
            height: 100%;
            padding: 0;
        }
        .calc {
            width: 100%;
            height: 100%;
            max-width: none;
            border-radius: 0;
        }
        #win-tool a {font-size: 1rem;line-height: 1.9rem;}
        #resize {
         left: 8px;
         font-size: 3.5rem;
         line-height:1.1rem !important;
         box-shadow: none !important;
         display: inline-block !important;}
        #bottom .row {margin: 0;}
        #bottom .row span {
            box-sizing: border-box;
            border-top: 0;
        }
        #bottom .row span.cheng {
         font-size: 2.1rem;
        }
        #bottom .row span.zero {
            width: 50.1%;
        }
    }
```

21.2.3 calc.js 文件

calc.js 文件是该 APP 中相应运算或数据交互的处理，具体请看下面的代码及注释。

```
    // app配置信息
    window.onload = function(){
        clickFunc();//点击功能
    };
    function clickFunc(){
         var container = document.getElementById("container");
        var calc = document.getElementById("calc"),
            spans =
document.getElementById("win-tool").getElementsByTagName("span"),
            equals = document.getElementById("equals"),     //等于符号
            remove = document.getElementById("remove");     //删除符号
    // 两个小按钮
    //  var close = document.getElementById("close"),       //关闭按钮
        var resultDiv = document.getElementById("result");  //结果区域
```

```
        // 历史记录
        var historyBox = document.getElementById("historyBox"),
            delBtn = historyBox.querySelector(".remove a");
        var historyUl = historyBox.querySelector("ul");
        // 关闭按钮
    close.onclick = function(e){
         var h = calc.offsetHeight + 15;
         calc.style.WebkitTransform = "translateY("+ h+"px)";
         calc.style.transform = "translateY("+ h+"px)";
         e.stopPropagation();
    };
        // 点击键盘
        var keyBorders = document.querySelectorAll("#bottom span"),
            express = document.getElementById("express"),//计算表达式
            res =  document.getElementById("res"),  //输出结果
            keyBorde = null;        //键盘
        var preKey = "",            //上一次按的键盘
            isFromHistory = false;  //是否来自历史记录
        //符号
        var symbol = {"+":"+","-":"-","×":"*","÷":"/","%":"%","=":"="};
        // 键盘按钮
        for(var j=0; j <keyBorders.length; j++){
            keyBorde = keyBorders[j];
            keyBorde.onclick = function() {
                var number = this.dataset["number"];
                clickNumber(number);
            };
        }
        // 点击键盘进行输入
        // @param {string} number 输入的内容
        function clickNumber(number){
         var resVal = res.innerHTML;     //结果
            var exp = express.innerHTML; //表达式
            //表达式最后一位的符号
            var expressEndSymbol = exp.substring(exp.length-1,exp.length);
            //点击的不是删除键和复位键时才能进行输入
            if(number !== "←" || number !== "C"){
              //是否已经存在点了，如果存在，就不能接着输入点号了,且上一个字符不是符号字符
              var hasPoint = (resVal.indexOf('.') !== -1)?true:false;
              if(hasPoint && number === '.'){
                  //上一个字符如果是符号，变成0.xxx形式
                  if(symbol[preKey]){
                      res.innerHTML = "0";
                  }else{
                      return false;
                  }
              }
                //转换显示符号
```

```
                if(isNaN(number)){
                    number = number.replace(/\*/g,"×").replace(/\//g,"÷");
                }
                //如果输入的都是数字，那么当输入达到固定长度时不能再输入了
                if(!symbol[number] && isResOverflow(resVal.length+1)){
                    return false;
                }
                //点击的是符号
                //计算上一次的结果
                if(symbol[number]){
                  //上一次点击的是不是符号键
                    if(preKey !== "=" && symbol[preKey]){
                        express.innerHTML = exp.slice(0,-1) + number;
                    }else{
                        if(exp == ""){
                            express.innerHTML = resVal + number;
                        }else{
                            express.innerHTML += resVal + number;
                        }
                        if(symbol[expressEndSymbol]){
                            exp =
express.innerHTML.replace(/×/g,"*").replace(/÷/,"/");
                            res.innerHTML = eval(exp.slice(0,-1));
                        }
                    }
                }else{
                    //如果首位是符号，0
                    if((symbol[number] || symbol[preKey] || resVal=="0") &&
number !== '.'){
                        res.innerHTML = "";
                    }
                    res.innerHTML += number;
                }
                preKey = number;
            }
        }
        // 相等，计算结果
        equals.onclick = function(){
            calcEques();
        };
        function calcEques(){
         var expVal = express.innerHTML, val = "";
            var resVal = res.innerHTML;
            //表达式最后一位的符号
            if(expVal){
                var expressEndSymbol =
expVal.substring(expVal.length-1,expVal.length);
                try{
```

```
                    if(!isFromHistory){
                        var temp = "";
                        if(symbol[expressEndSymbol] && resVal){
                            temp = expVal.replace(/×/g,"*").replace(/÷/,"/");
                            temp = eval(temp.slice(0,-1)) +
symbol[expressEndSymbol] + resVal;
                        }else{
                            temp = expVal.replace(/×/g,"*").replace(/÷/,"/");
                        }
                        val = eval(temp);
                    }else{
                        val = resVal;
                    }
                }catch(error){
                    val = "<span style='font-size:1em;color:red'>Erro: 计算出错!
</span>";
                }finally{
                    express.innerHTML = "";
                    res.innerHTML = val;
                    preKey = "=";
                    saveCalcHistory(expVal+resVal+"="+val);
                    isResOverflow(resVal.length);
                    isFromHistory = false;
                }
            }
        }
        // 移动端拨号功能
        //移动端长按事件
        $(equals).on("longTap",function(){
         //console.log("sdsdsd");
         var num = res.innerHTML;
            if(num && num !== "0"){
                  var regx = /^1[0-9]{2}[0-9]{8}$/;
                  if(regx.test(num)){
                      //console.log("是手机号码");
                      var telPhone = document.getElementById("telPhone");
                    telPhone.href = "tel:"+num;
                    telPhone.target = "_blank";
                    telPhone.click();
                  }
            }
        });
        // 复位操作
       var resetBtn = document.getElementById("reset");
        resetBtn.onclick = function(){
            res.innerHTML = "0";
            express.innerHTML = "";
        };
```

```
        // 减位操作
        remove.onclick = function(){
            var tempRes = res.innerHTML;
            if(tempRes.length>1){
                tempRes = tempRes.slice(0,-1);
                res.innerHTML = tempRes;
            }else{
                res.innerHTML = 0;
            }
        };
        // 历史功能
        var history = document.getElementById("history"),
            historyBox = document.getElementById("historyBox");
        var about = document.getElementById("about");
        about.onclick = history.onclick = function(e){
            e = e || window.event;
            var target = e.target.id || window.event.srcElement.id;
            historyBox.style.WebkitTransform = "none";
            historyBox.style.transform = "none";
            e.stopPropagation();
            //点击的是历史
            if(target == "h"){
             //恢复显示删除按钮
             delBtn.style.display = "inline-block";
                var keyArray = Mybry.wdb.getKeyArray();
                var separate = Mybry.wdb.constant.SEPARATE;
                keyArray.sort(function(a,b){
                    var n = a.split(separate)[1];
                    var m = b.split(separate)[1];
                    return m - n;
                });
                var html = [],val = "";
                for(var i=0; i<keyArray.length; i++){
                    val = Mybry.wdb.getItem(keyArray[i]);
                    html.push("<li>"+val+"</li>");
                }
                if(html.length>0){
                    historyUl.innerHTML = html.join("");
                }else{
                    historyUl.innerHTML = "尚无历史记录";
                }
                //把历史记录一条数据添加到计算器
                var hLis = historyUl.querySelectorAll("li");
                for(var i=0; i<hLis.length; i++){
                    hLis[i].onclick = function(){
                        var express = this.innerHTML;
                        var exp = express.split("=")[0],
                            res = express.split("=")[1];
```

```
                        resultDiv.querySelector("#express").innerHTML = exp;
                        resultDiv.querySelector("#res").innerHTML = res;
                        isFromHistory = true;
                    };
                }
            }
            if(target == "au"){
            //取消删除按钮显示
             delBtn.style.display = "none";
                historyBox.children[0].children[0].innerHTML = "<div
style='padding:5px;color:#000;'>"
                    + "<p><i class='iconfont'>&#xe60f;</i>纯CSS打造的一款计算器
</p>"
                    + "<p><i class='iconfont'>&#xe60f;</i>第一版，有待更新</p>"
                    + "</div>";
                //检查新版本
                updateapp();
            }
        };
        window.onclick = function(e){
            var e = e || window.event;
            var target = e.target.className || e.target.nodeName;
            var notTarget =
{"con":"con","remove":"remove","UL":"UL","P":"P"};
            if(!notTarget[target]){
                //如果设置了最小化
                var ts = historyBox.style.transform ||
historyBox.style.WebkitTransform;
                if(ts && ts == "none"){
                    historyBox.style.WebkitTransform = "translateY(102%)";
                    historyBox.style.transform = "translateY(102%)";
                }
            }
            //恢复显示删除按钮
            //historyBox.children[1].children[0].className = "icon del";
        };
        // 清空历史记录
        delBtn.onclick = function(e){
            var e = e || window.event;
            e.stopPropagation();
            if(Mybry.wdb.deleteItem("*")){
                historyUl.innerHTML = "尚无历史记录";
            }
        };
         // 保存计算历史记录
         // @param val要记录的表达式
        function saveCalcHistory(val){
            var key = Mybry.wdb.constant.TABLE_NAME +
```

```
Mybry.wdb.constant.SEPARATE + Mybry.wdb.getId();
            window.localStorage.setItem(key,val);
        }
        // 自动设置文字大小
        function isResOverflow(leng){
            var calc = document.getElementById("calc");
            var w = calc.style.width || getComputedStyle(calc).width ||
calc.currentStyle.width;
                w = parseInt(w);
            //判断是否是移动端
            if((Mybry.browser.versions.android ||
Mybry.browser.versions.iPhone || Mybry.browser.versions.iPad)
&& !symbol[preKey]) {
                if(leng > 15){
                    return true;
                }
            }else{
                if(leng > 10){
                    if(w == 300) {
                        maxCalc();
                    }else{
                        if(leng > 16){
                            return true;
                        }
                    }
                }
            }
            return false;
        }
    }
```

21.2.4 common.js 文件

common.js 文件是项目所需的工具类，具体请看下面的代码和注释。

```
    window.Mybry = {};
    Mybry.transitionEnd  = function(dom,callback){
         // 1.给谁加过渡结束事件
         // 2.过渡结束之后我们需要去做什么事情
        //基本的判断
        if(!dom || typeof dom != 'object' ) return false;
        dom.addEventListener('transitionEnd',function(){
            callback && callback();  //处理业务
        });
        dom.addEventListener('WebkitTransitionEnd',function(){
            callback && callback();  //处理业务
        });
    };
```

```
    //移动端单击事件
    Mybry.tap = function(dom,callback){
        //基本的判断
        if(!dom || typeof dom != 'object' ) return false;
        var startTime = 0,isMove = false;
        dom.addEventListener('touchstart',function(e){
            startTime = Date.now();
        });
        dom.addEventListener('touchmove',function(e){
            isMove = true;
        });
        dom.addEventListener('touchend',function(e){
            if((Date.now()-startTime) < 150 && !isMove){
                callback && callback(e);  //处理业务
            }
            startTime = 0,isMove = false;  //重置参数
        });
    };
    Mybry.wdb = {
        //自定义key的标识
        constant : {
            TABLE_NAME:"calc",     //表名称
            SEPARATE:"-"           //分隔符
        },
        //获取数据库最新的ID，递增
        getId : function(){
            var id = 0;  //key的索引
            var appDataKey = this.getKeyArray();
            var spearate = this.constant.SEPARATE;
            if(appDataKey.length>0){
                var indexArray = [];    //所有的索引值
                for(var i=0; i<appDataKey.length; i++){

indexArray.push(parseInt(appDataKey[i].split(spearate)[1]));
                }
                id = this. maxId(indexArray) + 1;
            }
            return id;
        },
        //获取单个数据，索引或key的名称
        getItem : function(value){
            if(!value) return false;
            if(isNaN(value)){
                return localStorage.getItem(value);
            }else{
                var key = localStorage.key(parseInt(value));
                return localStorage.getItem(key);
            }
```

```
    },
    deleteItem : function(value){
        if(!value) return false;
        if(isNaN(value)){
            //如果输入*号，删除所有数据
            if(value === "*"){
                var appDataKey = this.getKeyArray();
                for(var i=0; i<appDataKey.length; i++){
                    localStorage.removeItem(appDataKey[i]);
                }
            }else{
                localStorage.removeItem(value);
            }
        }else{
            var key = localStorage.key(parseInt(value));
            localStorage.removeItem(key);
        }
        return true;
    },
    _maxId : function(array){
        if(!array) return false;
        if(!Array.isArray(array)) return false;
        array.sort(function(a,b){
            return a - b;
        });
        return array[array.length-1];
    },
    getKeyArray : function(){
        var localStorage = window.localStorage;
        var storageLen = localStorage.length;
        var spearate = this.constant.SEPARATE,
            tableName = this.constant.TABLE NAME;
        //计算器所有的数据
        var appDataKey =[];
        if(storageLen>0){
            var itemKey = "";
            for(var i=0; i<storageLen; i++){
                //calc-0
                itemKey = localStorage.key(i);
                //判断是否是该应用的数据
                var flagIndex = itemKey.indexOf(spearate);
                if(flagIndex != -1 ){
                    var startWord = itemKey.split(spearate)[0];
                    if(startWord == tableName){
                        appDataKey.push(itemKey);
                    }
                }
            }
```

```
            }
            return appDataKey;
        }
    };
    Mybry.browser = {
        versions: function () {
            var u = navigator.userAgent,
                app = navigator.appVersion;
            return {
                //移动终端浏览器版本信息
                trident: u.indexOf('Trident') > -1, //IE内核
                presto: u.indexOf('Presto') > -1, //opera内核
                WebKit: u.indexOf('appleWebKit') > -1, //苹果、谷歌内核
                gecko: u.indexOf('Gecko') > -1 && u.indexOf('KHTML') == -1, //
火狐内核
                mobile: !!u.match(/appleWebKit.*Mobile.*/)
|| !!u.match(/appleWebKit/), //是否为移动终端
                ios: !!u.match(/\(i[^;]+;( U;)? CPU.+Mac OS X/), //ios终端
                android: u.indexOf('Android') > -1 || u.indexOf('Linux') > -1,
//android终端或uc浏览器
                iPhone: u.indexOf('iPhone') > -1 || u.indexOf('Mac') > -1, //
是否为iPhone或QQHD浏览器
                iPad: u.indexOf('iPad') > -1, //是否iPad
                Webapp: u.indexOf('Safari') == -1 //是否Web应该程序，没有头部与底部
            };
        }(),
        language: (navigator.browserLanguage ||
navigator.language).toLowerCase()
    }
```

第22章　开发求职招聘APP

本章将介绍一款求职招聘的手机网站，使用 jQuery Mobile 框架来制作。本项目设计以“绿色”为主色调，简约时尚。

22.1　项目概述

该手机网站项目主要包含登录、注册、个人中心、简历预览、简历编辑、投递记录、职位收藏、职位列表、职位详情等页面。

22.1.1　项目结构目录

本项目的目录结构如图 22-1 所示。

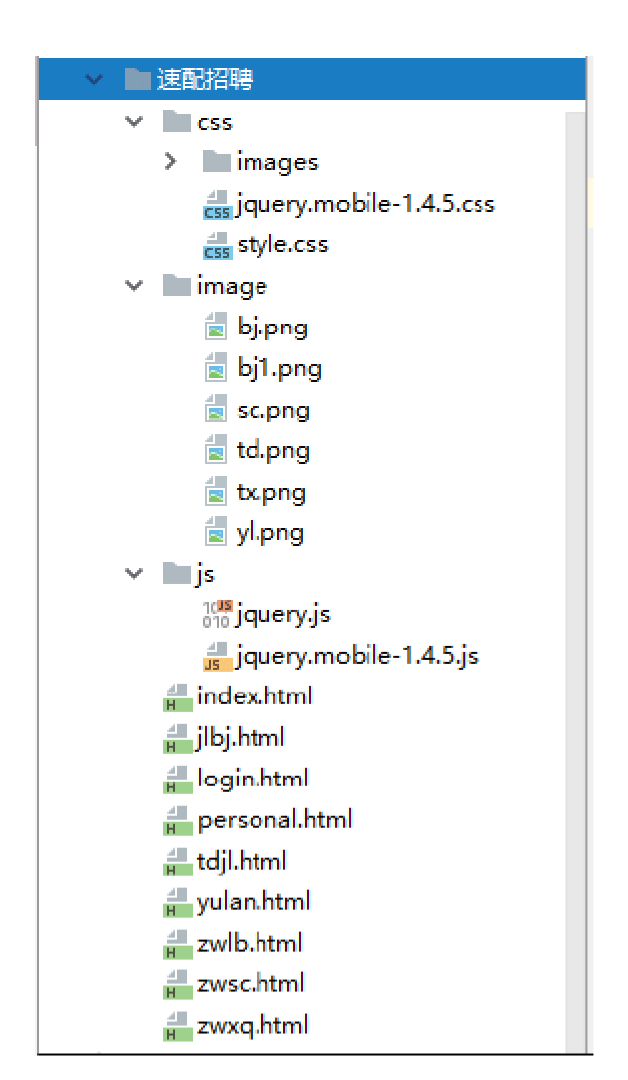


图 22-1　目录结构

具体内容如下所示。

（1）css 文件夹：包含了 jQuery Mobile 提供的图标（images）和 CSS 文件，以及自定义主题样式表 style.css。特别要注意，jQuery Mobile 提供的图标和 CSS 文件必须在同一个目录，

否则项目中使用的图标将不显示。

（2）image 文件夹：项目使用的图片。

（3）js 文件夹：包含 jQuery.js 和 jQuery Mobile.js 文件。

（4）index.html：登录页面。

（5）jlbj.html：简历编辑页面。

（6）login.html：注册页面。

（7）personal.html：个人中心页面。

（8）tdjl.html：投递简历页面。

（9）yulan.html：简历预览页面。

（10）zwlb.html：职位列表页面。

（11）zwsc.html：职位收藏页面。

（12）zwxq.html：职位详情页面。

22.1.2　项目效果展示

使用 Opera Mobile Emulator 模拟器来展示本项目效果。打开模拟器，选择“LG Optimus 3D”启动，把 index.html 文件拖入模拟器，页面效果如图 22-2 所示。

该页面是登录页面，用户进入可以选择登录或注册，当单击“免费注册”按钮时，将进入注册页面，效果如图 22-3 所示。在该页面编辑注册信息，选中“同意《我们的协议》”复选框，单击【免费注册】按钮，将返回登录页面，用户可以进行登录。



图 22-2　登录页面效果



图 22-3　注册页面效果

用户填好信息，单击“登录”按钮，将进入“个人中心”页面，效果如图 22-4 所示。单击“切换账号”按钮，将进入用户登录界面，可以选择注册或登录。

在“个人中心”页面中单击简历预览进入“简历预览”页面，效果如图 22-5 所示。在“简历预览”页面中单击“投递简历”按钮进入“职位列表”页面，效果如图 22-6 所示。在职位列表中可以搜索要查找的职位，如在搜索框中输入“java”，列表会筛选出与 java 相关的列表，如图 22-7 所示。



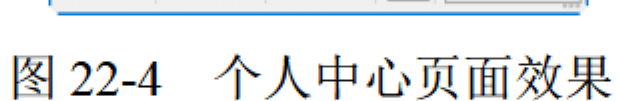

图 22-4　个人中心页面效果



图 22-5　简历预览页面效果

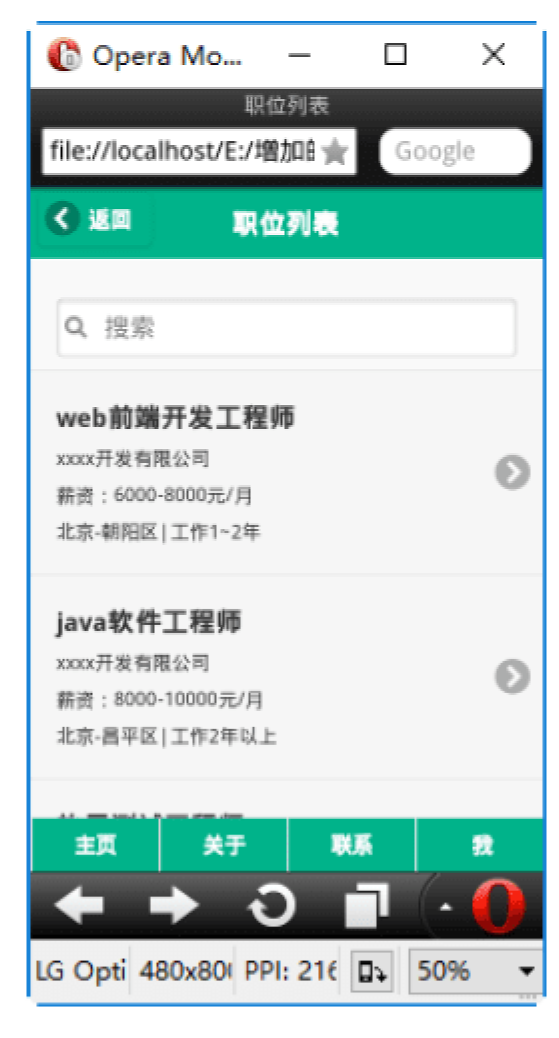


图 22-6　职位列表页面效果

在“个人中心”页面中单击简历编辑，进入“简历编辑”页面，把婚姻状况改成“已婚”，页面效果如图 22-8 和图 22-9 所示。单击“保存”按钮，页面将返回到“个人中心”页面。



图 22-7　简历编辑效果



图 22-8　简历编辑效果



图 22-9　简历编辑完成效果

在“个人中心”页面中单击投递记录，进入“投递记录”页面，效果如图 22-10 所示。单

击“投递记录”中的列表项目时，将进入“职位详情”页面。

在“个人中心”页面中单击职位收藏，进入“职位收藏”页面，效果如图 22-11 所示。单击“职位收藏”中的列表项目时，将进入“职位详情”页面，效果如图 22-12 所示。

在“职位列表”页面中单击“职位列表”中的列表项目时，将进入“职位详情”页面。

在“职位详情”页面中单击“投递简历”按钮，进入“职位列表”页面。



图 22-10　“投递记录”页面效果

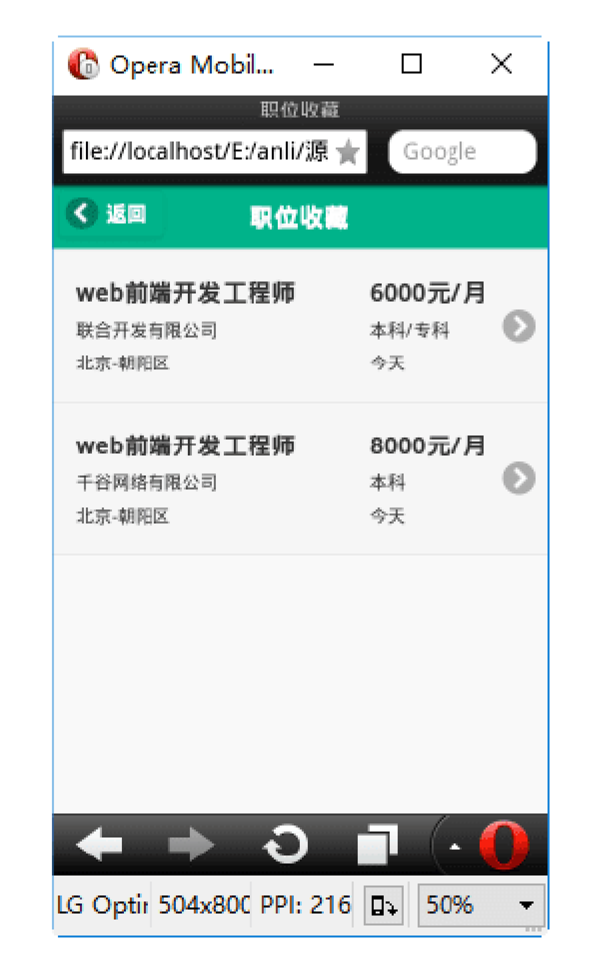


图 22-11　“职位收藏”页面效果



图 22-12　“职位详情”页面效果

22.2　项目设计

本项目是基于 jQuery Mobile 框架来制作的，首先引入 jQuery Mobile 框架的文件，然后引入 jQuery 框架的文件及 style.css 文件，具体如下面的代码所示。

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/jquery.mobile-1.4.5.css">
<!--自定义的主题-->
<link rel="stylesheet" href="css/style.css">
<script src="js/jquery.js"></script>
<script src="js/jquery.mobile-1.4.5.js"></script>
```

本项目中的每个页面都必须引入上面的文件，在后面的介绍中将不再赘述。

其中引入的 style.css 文件是自定义的主题，在本项目中主要用于头部工具栏（header），具体代码如下所示。

```
.ui-bar-x{
   background:#00B38A;
   color: #fff;
}
```

22.2.1 设计登录和注册页面

登录和注册页面是一个网站中必不可少的，本节我们使用 jQuery Mobile 框架的内容来制作。

1. 登录页面

index.html 是登录页面。

在头部栏中添加主题“x”。在内容区域中，首先是一个<form>表单，用于用户输入登录信息，接下来使用了两列网格布局，左边是对于忘记密码的用户提供帮助，右边可以选择是否勾选，勾选下次将自动登录。再往下是“登录”按钮和进入“免费注册”页面的按钮。注意 jQuery Mobile 框架中，超链接<a>的 href 属性指向其他文件时，需要添加“data-ajax=false”属性类别。

具体请参考下面的代码。

```
    <body>
    <div data-role="page">
        <!--data-position="fixed"属性类别用于固定头部和尾部的位置，data-theme设置自定义的主题-->
        <div data-role="header" data-position="fixed" data-theme="x">
            <h1>登录</h1>
        </div>
        <div data-role="content">
            <!--添加姓名和密码输入框-->
            <form>
                <label for="name">姓名：</label>
                <input type="text" name="name" id="name" placeholder="请输入用户名称"/>
                <label for="password">密码：</label>
                <input type="password" name="password" id="password"
placeholder="请输入密码"/>
            </form><br>
            <!--使用jquery mobile网格系统中的2列布局-->
            <div class="ui-grid-a" style="text-align: center;">
                <div class="ui-block-a">
                    <a href="#" style="font-size: 14px;">忘记密码?</a>
                </div>
                <div class="ui-block-b">
                    <input type="checkbox" name="checkbox" id="checkbox"/>
                    <label for="checkbox" style="border: 0px;font-size:
14px;margin:-15px;">下次自动登录</label>
                </div>
            </div><br>
            <a href="personal.html" data-ajax="false" data-role="button"
style="background:#00B38A;color:white;">登录</a>
            <p style="text-align: center">-----------------还没有账号
------------------</p>
            <a href="login.html" data-ajax="false" data-role="button"
```

```
style="color:#71BC44;">免费注册</a>
        </div>
    </div>
    </body>
```

2. 注册页面

login.html 是注册页面。

在头部栏中，首先添加自定义主题“x”，然后添加一个返回的按钮，添加 onClick="javascript:history.back(-1);"属性类别，用于返回上一个页面。在内容区域中，先是一个<form>表单用于填写注册信息，比登录页面多了一个邮箱信息，紧接着是一个复选框，只有当用户勾选了以后才可以注册，最后是免费注册按钮，并添加了相应的 CSS 样式，来使整个页面更协调。

具体请参考下面的代码。

```
    <body>
    <div data-role="page">
        <div data-role="header" data-position="fixed" data-theme="x">
            <a href="javascript :;" onClick="javascript :history.back(-1);"
            <a href="javascript :;" onClick="javascript :history.back(-1);"
data-icon="carat-l" style="background:#00B38A;color:white;border: 0;">返回
</a>
            <h1>注册</h1>
        </div>
        <div data-role="content">
            <form>
                <label for="name">姓名：</label>
                <input type="text" name="name" id="name" placeholder="请输入用户
名称"/>
                <label for="password">密码：</label>
                <input type="password" name="password" id="password"
placeholder="请输入密码"/>
                <label for="email">邮箱：</label>
                <input type="email" name="email" id="email" placeholder="请输入
邮箱"/>
            </form>
            <input type="checkbox" name="checkbox" id="checkbox"/>
            <label for="checkbox" style="border: 0px;font-size: 14px;">同意<a>
《我们的协议》</a></label>
            <a href="index.html" data-ajax="false" data-role="button"
style="background:#00B38A;color:white;">免费注册</a>
        </div>
    </div>
    </body>
```

22.2.2 设计个人中心页面

personal.html 是个人中心页面。

在头部栏中，先是添加自定义主题“x”，然后添加一个返回的按钮，用于返回上一个页面，接着添加一个“切换账号”按钮，用于用户退出登录或重新注册登录。

在内容区域中，首先设置用户信息展示的区域，然后设置用户照片的区域，用户可以选择上传，照片下面是登录用户的姓名，再往下是该用户最后登录的信息。在用户信息展示区域下面紧接着是一个列表，登录后可以进行一系列的操作。

在底部栏中添加了一个导航条，这里我们主要用来进行页面之间的跳转和展示该网站 APP 的一些信息。

具体请参考下面的代码。

```
    <style>
    .col li a{
         background:#00B38A!important;
         color: white!important;
    }
    </style>
    <body>
    <div data-role="page">
        <div data-role="header" data-position="fixed" data-theme="x">
            <a href="javascript :;" onClick="javascript :history.back(-1);"
data-icon="carat-l" style="background:#00B38A;color:white;border: 0;">返回
</a>
            <h1>个人中心</h1>
            <a href="index.html" data-ajax="false"
style="background:#00B38A;color:white;">切换账号</a>
        </div>
        <div data-role="content">
            <div style="background-image: url('images/bj.png');text-align:
center;">
                <br><div><img src="images/tx.png" width="80px"/></div>
                <div style="margin: 5px ">林欢欢</div>
                <div style="font-size: 0.7em;color: #999;">
                    最后登录时间：2018.10.30 最后更新时间：2018.10.29
                </div>
            </div>

            <ul data-role="listview">
                <li>
                    <a href="yulan.html" data-ajax="false">
                        <img src="images/yl.png" class="ui-li-icon">
                        简历预览
                    </a>
```

```
            </li>
            <li>
                <a href="jlbj.html" data-ajax="false">
                    <img src="images/bj1.png" class="ui-li-icon">
                    简历编辑
                </a>
            </li>
            <li>
                <a href="tdjl.html" data-ajax="false">
                    <img src="images/td.png" class="ui-li-icon">
                    投递记录
                    <!--设置气泡数字-->
                    <span class="ui-li-count">4</span>
                </a>
            </li>
            <li>
                <a href="zwsc.html" data-ajax="false">
                    <img src="images/sc.png" class="ui-li-icon">
                    职位收藏
                    <span class="ui-li-count">2</span>
                </a>
            </li>
        </ul>
    </div>
    <div data-role="footer" data-position="fixed">
        <!--data-role="navbar"用于添加导航-->
        <div data-role="navbar" data-theme="b">
            <ul  class="col">
                <li><a href="zwlb.html" data-ajax="false">主页</a></li>
                <li><a href="#">关于</a></li>
                <li><a href="#">联系</a></li>
                <li><a href="personal.html" data-ajax="false">我</a></li>
            </ul>
        </div>
    </div>
</div>
</body>
```

22.2.3 设计简历预览页面

yulan.html 是简历预览页面。

在头部栏中，先是添加自定义主题“x”，然后添加一个返回的按钮，用于返回上一个页面。

在内容区域中，我们使用一个表格来设计布局，左边是问题，右边是答案。

在底部栏中，先是添加自定义主题“x”，然后添加一个“投递简历”按钮并设置相关的样式。

具体请参考下面的代码。

```
    <body>
    <div data-role="page">
        <div data-role="header" data-position="fixed" data-theme="x">
            <a href="javascript :;" onClick="javascript :history.back(-1);"
data-icon="carat-l" style="background:#00B38A;color:white;border: 0;">返回
</a>
            <h1>简历预览</h1>
        </div>
        <div data-role="content">
            <h3  style="border-bottom: 2px solid #999; padding: 8px;">
                <span style="border-left: 8px solid lawngreen;padding-left:
10px;">基本信息</span>
            </h3>
            <!--这里我们使用表格来进行页面布局-->
            <table>
                <tr><td>姓<i style="width: 2em;display: inline-block;"></i>名:
</td><td>林欢欢</td></tr>
                <tr><td>性<i style="width: 2em;display: inline-block;"></i>别:
</td><td>女</td></tr>
                <tr><td>出生日期: </td><td>1995年11月8日</td></tr>
                <tr><td>婚姻状况: </td><td>未婚</td></tr>
                <tr><td>民<i style="width: 2em;display: inline-block;"></i>族:
</td><td>汉</td></tr>
                <tr><td>身<i style="width: 2em;display: inline-block;"></i>高:
</td><td>172cm</td></tr>
                <tr><td>毕业院校</td><td>北京清华大学</td></tr>
                <tr><td>专<i style="width: 2em;display: inline-block;"></i>业:
</td><td>土木工程</td></tr>
                <tr><td>最高学历: </td><td>本科</td></tr>
                <tr><td>手<i style="width: 2em;display: inline-block;"></i>机:
</td><td>12344445555</td></tr>
                <tr><td>现<i style="width: 2em;display: inline-block;"></i>住:
</td><td>北京海淀区xxxx</td></tr>
                <tr><td>籍<i style="width: 2em;display: inline-block;"></i>贯:
</td><td>北京朝阳区xxxx</td></tr>
            </table>
        </div>
        <div data-role="footer" data-position="fixed" data-theme="x">
            <a href="zwlb.html" data-ajax="false" type="button" style="width:
100%;height: 100%;background:#00B38A;color: white;border: 0;">投递简历</a>
        </div>
    </div>
    </body>
```

22.2.4 设计简历编辑页面

jlbj.html 是简历编辑页面。

在头部栏中，先是添加自定义主题“x”，然后添加一个返回的按钮，用于返回上一个页面。

在内容区域中，使用一个表格来设计布局，左边是问题，右边是<input>文本框，用户可以修改相应的信息。

在底部栏中，先是添加自定义主题“x”，然后添加一个“保存”按钮并设置相关的样式。

具体请参考下面的代码。

```
    <body>
    <div data-role="page">
        <div data-role="header" data-position="fixed" data-theme="x">
            <a href="javascript :;" onClick="javascript :history.back(-1);"
data-icon="carat-l" style="background:#00B38A;color:white;border: 0;">返回
</a>
            <h1>简历编辑</h1>
        </div>
        <div data-role="content">
            <h3 style="border-bottom: 2px solid #999; padding: 8px;">
                <span style="border-left: 8px solid lawngreen;padding-left:
10px;">基本信息（必填）</span>
            </h3>
            <!--这里我们使用表格来进行页面布局-->
            <table>
                <!--为了使页面更美观，添加了<i>标签，设置它的宽度为两个字符-->
                <tr><td>姓<i style="width: 2em;display: inline-block;"></i>名:
</td><td><input type="text" placeholder="林欢欢"></td></tr>
                <tr><td>性<i style="width: 2em;display: inline-block;"></i>别:
</td><td><input type="text" placeholder="女"></td></tr>
                <tr><td>出生日期：</td><td><input type="text" placeholder="1995
年11月8日"></td></tr>
                <tr><td>婚姻状况：</td><td><input type="text" placeholder="未婚
"></td></tr>
                <tr><td>民<i style="width: 2em;display: inline-block;"></i>族:
</td><td><input type="text" placeholder="汉"></td></tr>
                <tr><td>身<i style="width: 2em;display: inline-block;"></i>高:
</td><td><input type="text" placeholder="172cm"></td></tr>
                <tr><td>毕业院校</td><td><input type="text" placeholder="北京清华
大学"></td></tr>
                <tr><td>专<i style="width: 2em;display: inline-block;"></i>业:
</td><td><input type="text" placeholder="土木工程"></td></tr>
                <tr><td>最高学历：</td><td><input type="text" placeholder="本科
"></td></tr>
                <tr><td>手<i style="width: 2em;display: inline-block;"></i>机:
```

```
</td><td><input type="text" placeholder="12344445555"></td></tr>
              <tr><td>现<i style="width: 2em;display: inline-block;"></i>住:
</td><td><input type="text" placeholder="北京海淀区xxxx"></td></tr>
              <tr><td>籍<i style="width: 2em;display: inline-block;"></i>贯:
</td><td><input type="text" placeholder="北京朝阳区xxxx"></td></tr>
          </table>
       </div>
       <div data-role="footer" data-position="fixed" data-theme="x">
          <a href="personal.html" data-ajax="false" type="button"
style="width: 100%;height: 100%;background:#00B38A;color: white;border: 0;">
保存</a>
       </div>
    </div>
    </body>
```

22.2.5　设计投递记录和职位收藏页面

投递记录和职位收藏页面设计布局是一样的。

1. 投递记录页面

tdjl.html 是投递记录页面。

在头部栏中，先是添加自定义主题“x”，然后添加一个返回的按钮，用于返回上一个页面。

在内容区域中，我们使用一个列表来设计，在每一个列表项中添加了两个<div>并设置浮动样式，分别在两个<div>中填写相应的信息。

具体请参考下面的代码。

```
    <style>
        .left{float: left;}
        .right{float: right;}
    </style>
    <body>
    <div data-role="page">
       <div data-role="header" data-position="fixed" data-theme="x">
          <a href="javascript :;" onClick="javascript :history.back(-1);"
data-icon="carat-l" style="background:#00B38A;color:white;border: 0;">返回
</a>
          <h1>投递记录</h1>
       </div>
       <div data-role="content">
          <ul data-role="listview">
             <li>
                <a href="zwxq.html" data-ajax="false">
                   <div class="left">
                      <h3>Web前端开发工程师</h3>
```

```
                    <p>联合开发有限公司</p>
                    <p>北京-朝阳区</p>
                </div>
                <div class="right">
                    <h3>6000元/月</h3>
                    <p>本科/专科</p>
                    <p>今天</p>
                </div>
            </a>
        </li>
        <li>
            <a href="zwxq.html" data-ajax="false">
                <div class="left">
                    <h3>Web前端开发工程师</h3>
                    <p>千谷网络有限公司</p>
                    <p>北京-朝阳区</p>
                </div>
                <div class="right">
                    <h3>8000元/月</h3>
                    <p>本科</p>
                    <p>今天</p>
                </div>
            </a>
        </li>
        <li>
            <a href="zwxq.html" data-ajax="false">
                <div class="left">
                    <h3>Web前端开发工程师</h3>
                    <p>宏伟网络有限公司</p>
                    <p>北京-海淀区</p>
                </div>
                <div class="right">
                    <h3>8000元/月</h3>
                    <p>本科</p>
                    <p>8-25</p>
                </div>
            </a>
        </li>
        <li>
            <a href="zwxq.html" data-ajax="false">
                <div class="left">
                    <h3>Web前端开发工程师</h3>
                    <p>飞驰网络有限公司</p>
                    <p>北京-昌平区</p>
                </div>
                <div class="right">
                    <h3>8000元/月</h3>
                    <p>本科</p>
```

```
                    <p>8-20</p>
                </div>
            </a>
        </li>
    </ul>
    </div>
</div>
</body>
```

2. 职位收藏页面

zwsc.html 是职位收藏页面。

职位收藏页面与投递记录页面基本一致，这里就再不赘述了。

具体请参考下面的代码。

```
<style>
    .left{float: left;}
    .right{float: right;}
</style>
<body>
<div data-role="page">
    <div data-role="header" data-position="fixed" data-theme="x">
        <a href="javascript :;" onClick="javascript :history.back(-1);"
data-icon="carat-l" style="background:#00B38A;color:white;border: 0;">返回
</a>
        <h1>职位收藏</h1>
    </div>
    <div data-role="content">
        <ul data-role="listview">
            <li>
                <a href="zwxq.html" data-ajax="false">
                    <div class="left">
                        <h3>Web前端开发工程师</h3>
                        <p>联合开发有限公司</p>
                        <p>北京-朝阳区</p>
                    </div>
                    <div class="right">
                        <h3>6000元/月</h3>
                        <p>本科/专科</p>
                        <p>今天</p>
                    </div>
                </a>
            </li>
            <li>
                <a href="zwxq.html" data-ajax="false">
                    <div class="left">
                        <h3>Web前端开发工程师</h3>
                        <p>千谷网络有限公司</p>
```

```
                    <p>北京-朝阳区</p>
                </div>
                <div class="right">
                    <h3>8000元/月</h3>
                    <p>本科</p>
                    <p>今天</p>
                </div>
            </a>
        </li>
    </ul>
  </div>
</div>
</body>
```

22.2.6　设计职位列表页面

zwlb.html 是职位列表页面。

在头部栏中，先是添加自定义主题“x”，然后添加一个返回的按钮，用于返回上一个页面。

在内容区域中，首先使用一个表格来布局，左边是<input>搜索框，右边是搜索按钮。

接下来是一个列表，用来展示职位信息。

在底部栏中添加了一个导航条，这里我们主要用来进行页面之间的跳转和展示该网站 APP 的一些信息。

具体请参考下面的代码。

```
<style>
    .col li a{
        background:#00B38A!important;
        color: white!important;
    }
</style>
<body>
<div data-role="page">
    <div data-role="header" data-position="fixed" data-theme="x">
        <a href="javascript :;" onClick="javascript :history.back(-1);"
data-icon="carat-l" style="background:#00B38A;color:white;border: 0;">返回
</a>
        <h1>职位列表</h1>
    </div>
    <div data-role="content">
        <!--这里我们使用了搜索字段，在输入框中输入想要搜索的职位，可以筛选出对应的职位，
要想实现效果需要以下几个步骤-->
        <!--1.给要过滤的元素添加data-filter="true"属性-->
        <!--2.创建<input>元素并指定id，在该元素上加上data-type="search"属性，这
样就能创建基本的搜索字段。将<input>元素放置于一个表单中，表单 <form> 元素使用
```

```
"ui-filterable"类，该类会调整搜索字段与过滤元素的外边距-->
          <!--3.接着为过滤的元素添加data-input属性，属性值是<input>元素的id-->
          <form class="ui-filterable">
             <input id="myFilter" data-type="search" placeholder="搜索">
          </form>
          <ul data-role="listview" data-filter="true" data-input="#myFilter"
data-autodividers="true" data-inset="true">
             <li>
                <a href="zwxq.html" data-ajax="false">
                   <h3>Web前端开发工程师</h3>
                   <p>xxxx开发有限公司</p>
                   <p>薪资：6000-8000元/月</p>
                   <p>北京-朝阳区|工作1~2年</p>
                </a>
             </li>
             <li>
                <a href="zwxq.html" data-ajax="false">
                   <h3>java软件工程师</h3>
                   <p>xxxx开发有限公司</p>
                   <p>薪资：8000-10000元/月</p>
                   <p>北京-昌平区|工作2年以上</p>
                </a>
             </li>
             <li>
                <a href="zwxq.html" data-ajax="false">
                   <h3>软件测试工程师</h3>
                   <p>xxxx开发有限公司</p>
                   <p>薪资：4000-6000元/月</p>
                   <p>北京-朝阳区|工作1~2年</p>
                </a>
             </li>
             <li>
                <a href="zwxq.html" data-ajax="false">
                   <h3>Web前端开发工程师</h3>
                   <p>xxxx开发有限公司</p>
                   <p>薪资：10000-15000元/月</p>
                   <p>北京-朝阳区|工作3年以上</p>
                </a>
             </li>
          </ul>
       </div>
       <div data-role="footer" data-position="fixed" >
          <div data-role="navbar" data-theme="b">
             <ul  class="col">
                <li><a href="zwlb.html" data-ajax="false">主页</a></li>
                <li><a href="#">关于</a></li>
                <li><a href="#">联系</a></li>
                <li><a href="personal.html" data-ajax="false">我</a></li>
```

```
            </ul>
         </div>
      </div>
   </div>
   </body>
```

22.2.7　设计职位详情页面

zwxq.html 是职位详情页面。

在头部栏中，先是添加自定义主题“x”，然后添加一个返回的按钮，用于返回上一个页面。具体请参考下面的代码。

```
   <body>
   <div data-role="page">
      <div data-role="header" data-position="fixed" data-theme="x">
         <a href="javascript :;" onClick="javascript :history.back(-1);"
data-icon="carat-l" style="background:#00B38A;color:white;border: 0;">返回
</a>
         <h1>职位详情</h1>
      </div>
      <div data-role="content">
         <div>
            <h2 style="border-bottom: 2px solid #999; padding: 8px;">
               <a style="border-left: 8px solid lawngreen;padding-left:
10px;">职位名称</a>
            </h2>
            <p>地区：<span>北京-朝阳区</span></p>
            <p>时间：2018-7-20</p>
            <p>薪资：6000-8000元/月</p>
         </div>
         <h3 style="border-bottom: 2px solid #999; padding: 8px;">
            <a style="border-left: 8px solid lawngreen;padding-left:
10px;">xxxx网络开发有限公司</a>
         </h3>
         <div>
            <p>公司简介：xxxx</p>
            <p>公司规模：xxxx</p>
            <p>发展方向：xxxx</p>
         </div>
      </div>
      <div data-role="footer" data-position="fixed" data-theme="x">
         <a href="zwlb.html" data-ajax="false" type="button" style="width:
100%;height: 100%;background:#00B38A;color: white;border: 0;">投递简历</a>
      </div>
   </div>
   </body>
```

22.3　项目打包成 APP

项目打包使用 HBuilder 工具。HBuilder 默认是在云端打包的，也就是将代码提交上去进行打包，然后下载打好的包。优点是不管机器配置的高低，只要网速快就可以很快打好包，当然也可以进行本地打包，那样就需要 Android 环境和 IOS 环境，不做推荐。

首先打开 HBuilder 工具，执行“文件”>>“新建”>>“移动 APP”命令，弹出新建项目界面，输入应用名称，位置可以根据需要自己选择，“选择模板”建议选择空模板，如图 22-13 所示。

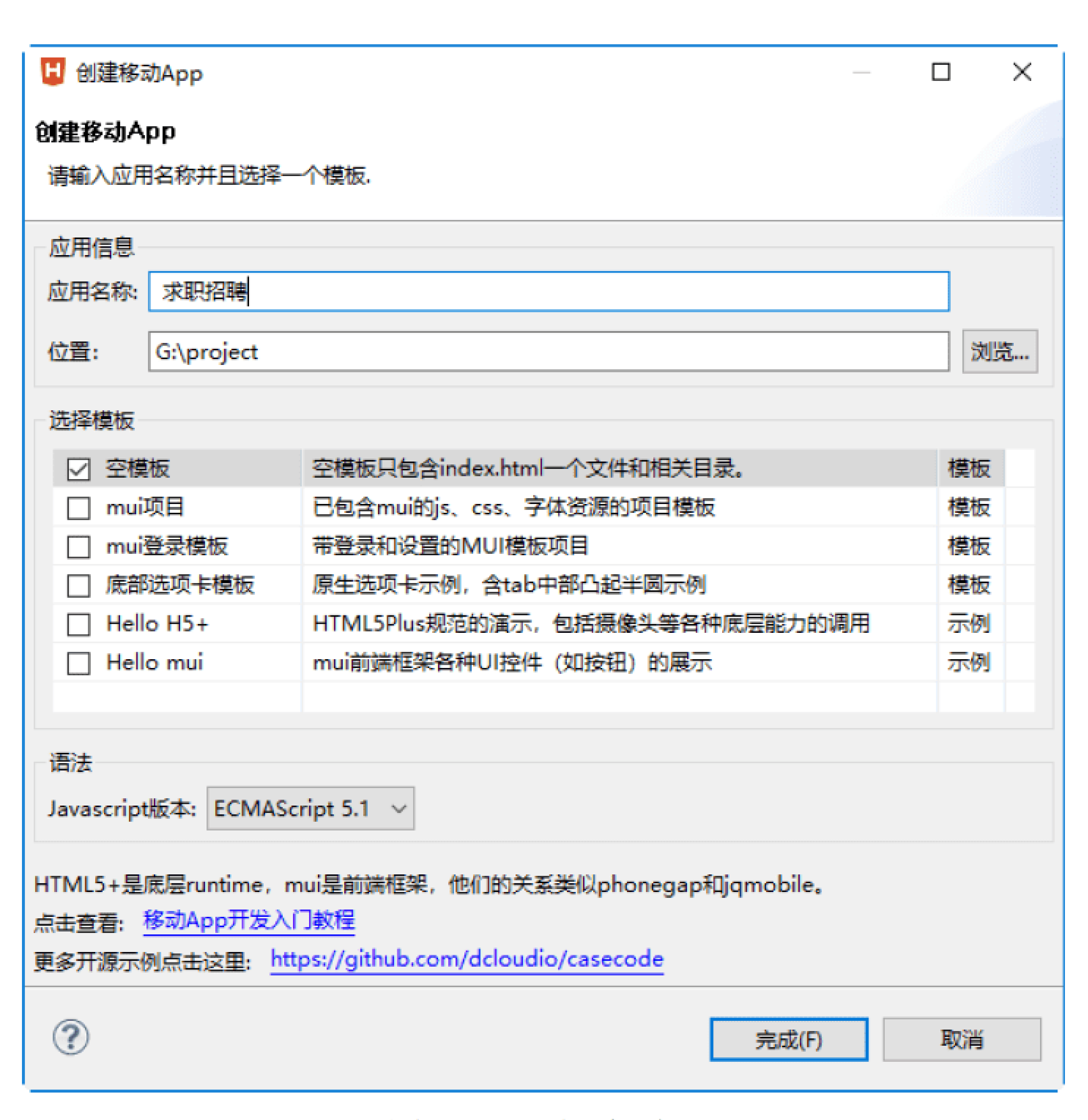


图 22-13　新建项目

新建完成后，在“项目管理器”会显示新建的项目目录，如图 22-14 所示。其中 css 文件夹、img 文件夹、js 文件夹和 index.html 文件都可以删除、替换或更改。unpackage 文件夹是放置 APP 图标和启动界面的图片；manifest.json 文件是移动 APP 的配置文件，用于指定应用的显示名称、图标、应用入口文件地址及需要使用的设备权限等信息，用户可通过 HBuilder 的可视化界面视图或源代码视图来配置移动 APP 的信息。

如果删除了 css 文件夹、img 文件夹、js 文件夹和 index.html 文件，就把自己的项目文件复制到新建的项目中，注意 html 文件中的引用路径需要保持正确，如图 22-15 所示。

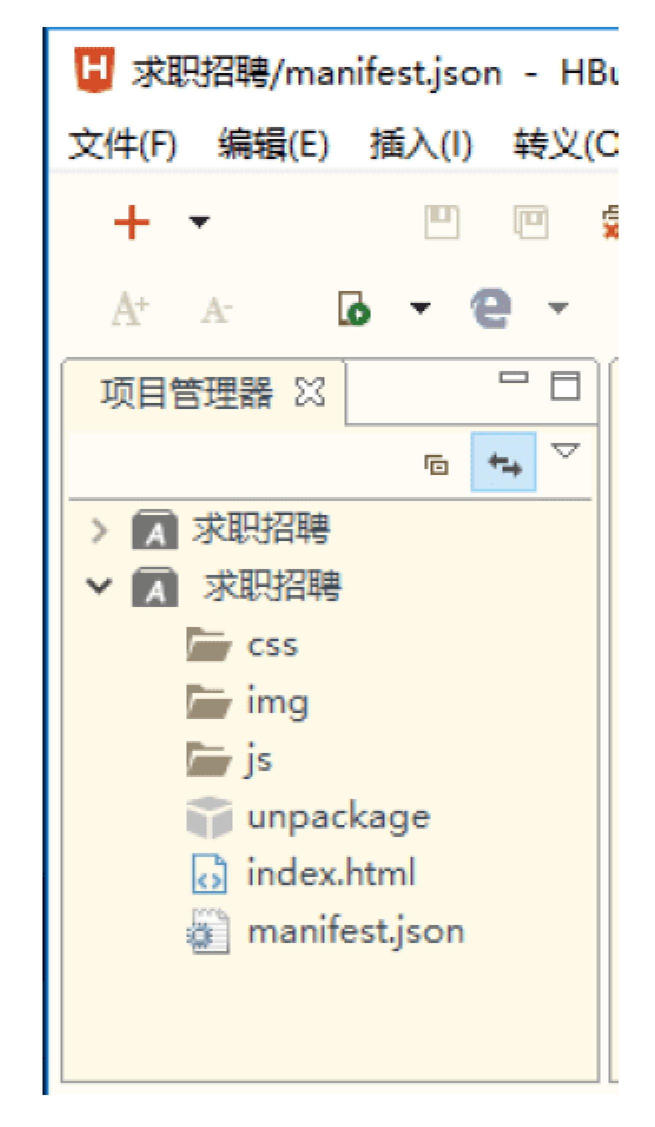


图 22-14　新建项目的目录

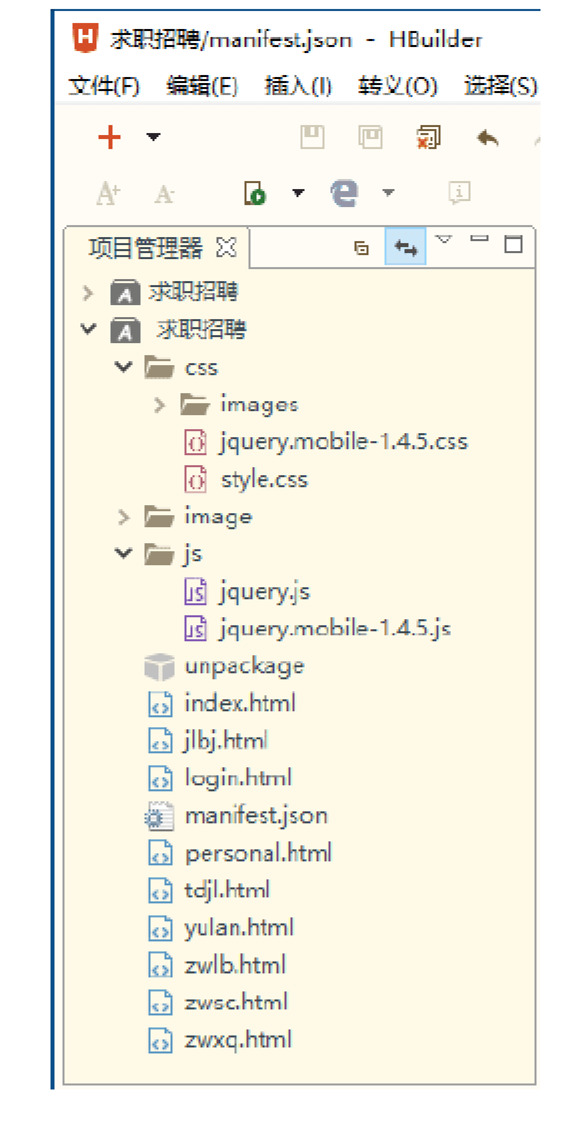


图 22-15　复制后的项目目录

文件复制完成后，双击打开 manifest.json 文件来配置 APP，如图 22-16 所示。这里全部保持默认就可以了。

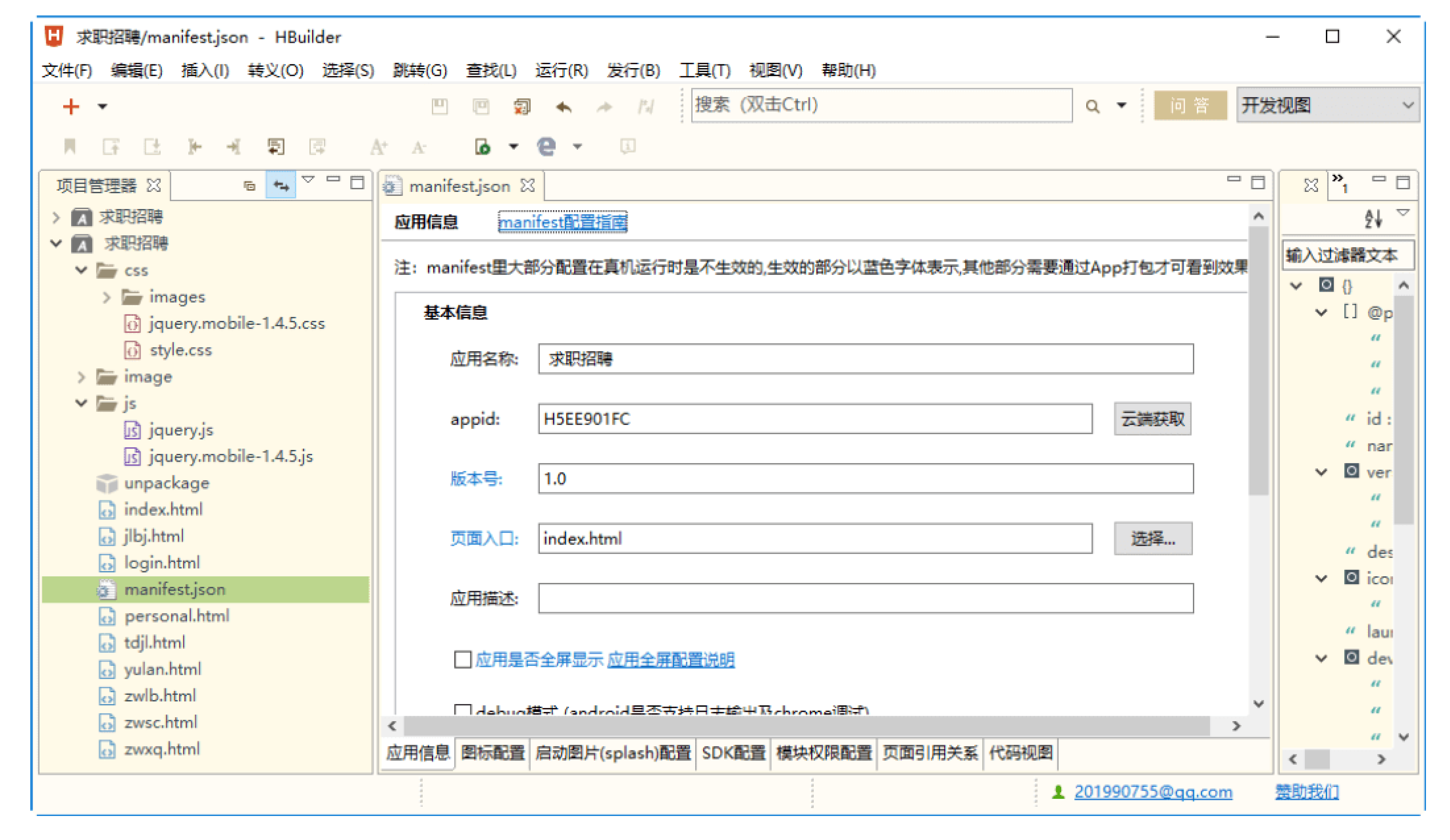


图 22-16　配置 APP 界面

接下来，在 HBuilder 中选择“发行”>>“云打包-打原生安装包”命令，如图 21-17 所示。

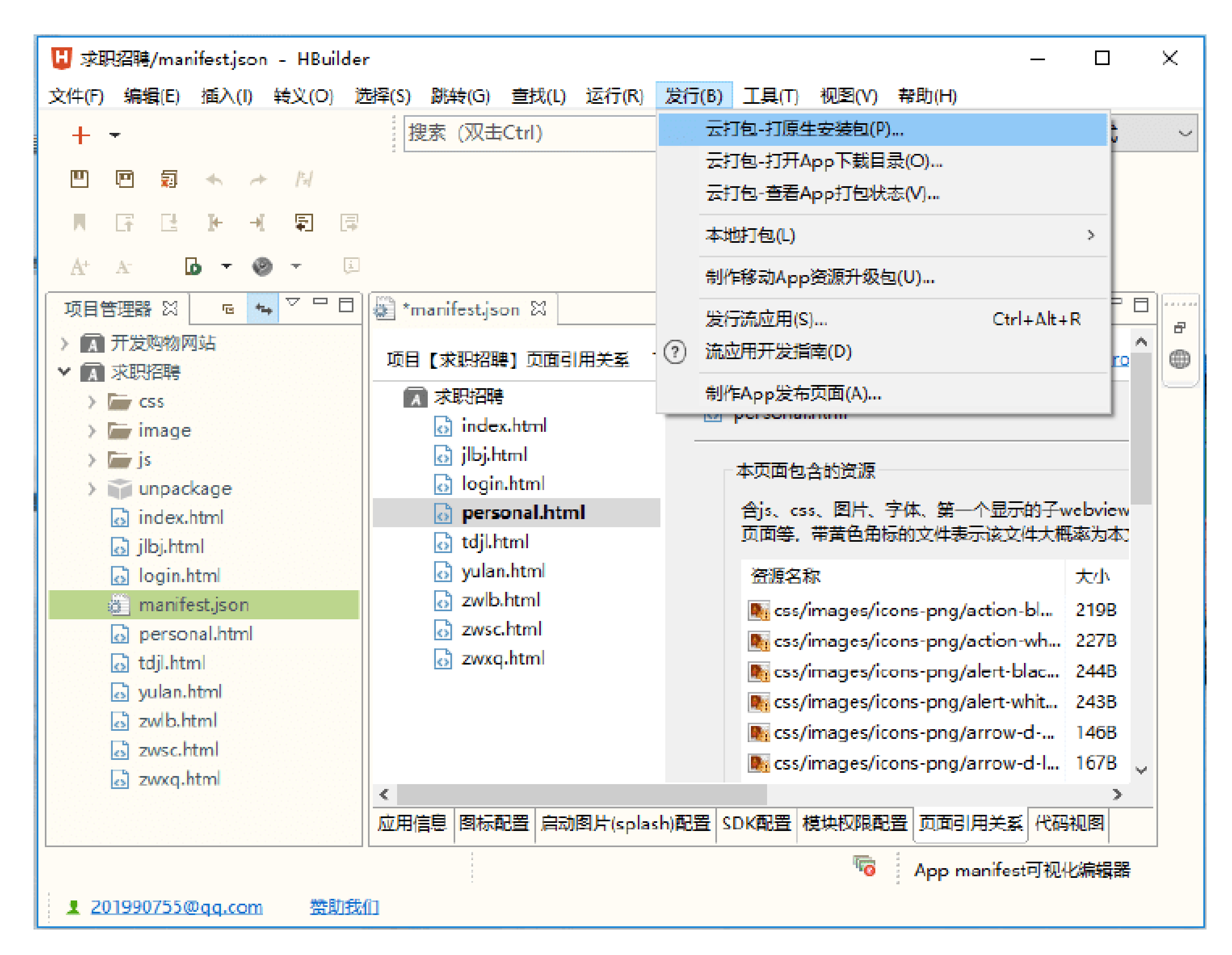


图 22-17 云打包

弹出如图 22-18 所示的页面，选中“Android”复选框，然后单击“打包”按钮进入打包界面。



图 22-18 打包

打包成功后，如图 22-19 所示。单击“手动下载”按钮下载到本地，下载完成后就可以安装到手机中运行了。



图 22-19　打包成功

在手机上安装成功后，打开 APP，页面效果与前面项目效果展示基本一致，效果如图 22-20~图 22-22 所示。



图 22-20　登录页面效果



图 22-21　个人中心页面效果



图 22-22　投递记录页面效果

第23章　项目实训3——开发购物网站APP

本章将介绍一个网上购物的网站，网站以 Bootstrap 框架技术为主，利用 Bootstrap 技术特点来实现响应式布局，可以在不同分辨率的设备上自适应显示。该网站页面设计风格简洁、大气，完美地诠释了 Bootstrap 框架的基本风格特点。

23.1　项目概述

该网站主要销售蔬菜、水果和干果等产品。具体功能将在下面的小节中介绍。

23.1.1　项目结构目录

本项目的目录结构如图 23-1 所示。

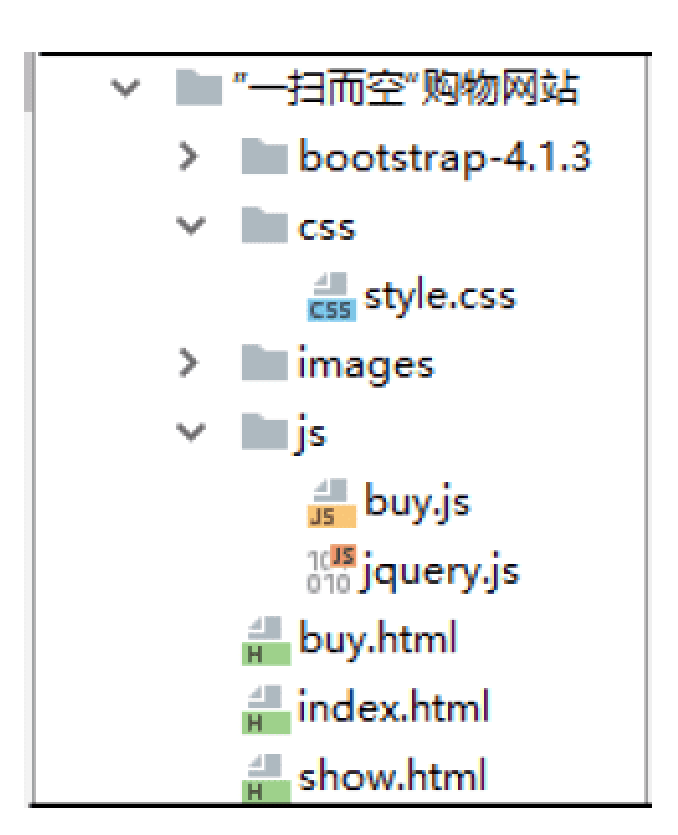


图 23-1　目录结构

具体内容如下。

（1）Bootstrap-4.1.3 文件夹：这里包含 Bootstrap 框架的最新版本。

（2）css 文件夹：项目的 CSS 样式文件。

（3）images 文件夹：项目使用的图片。

（4）js 文件夹：包含 jQuery.js 和项目的 JS 文件。

（5）buy.html：购买页面。

（6）index.html：项目的首页。

（7）show.html：更多展示页面。

23.1.2　项目效果展示

下面使用 IE 11.0 浏览器来展示项目效果。首先打开 index.html 页面，效果如图 23-2 所示。

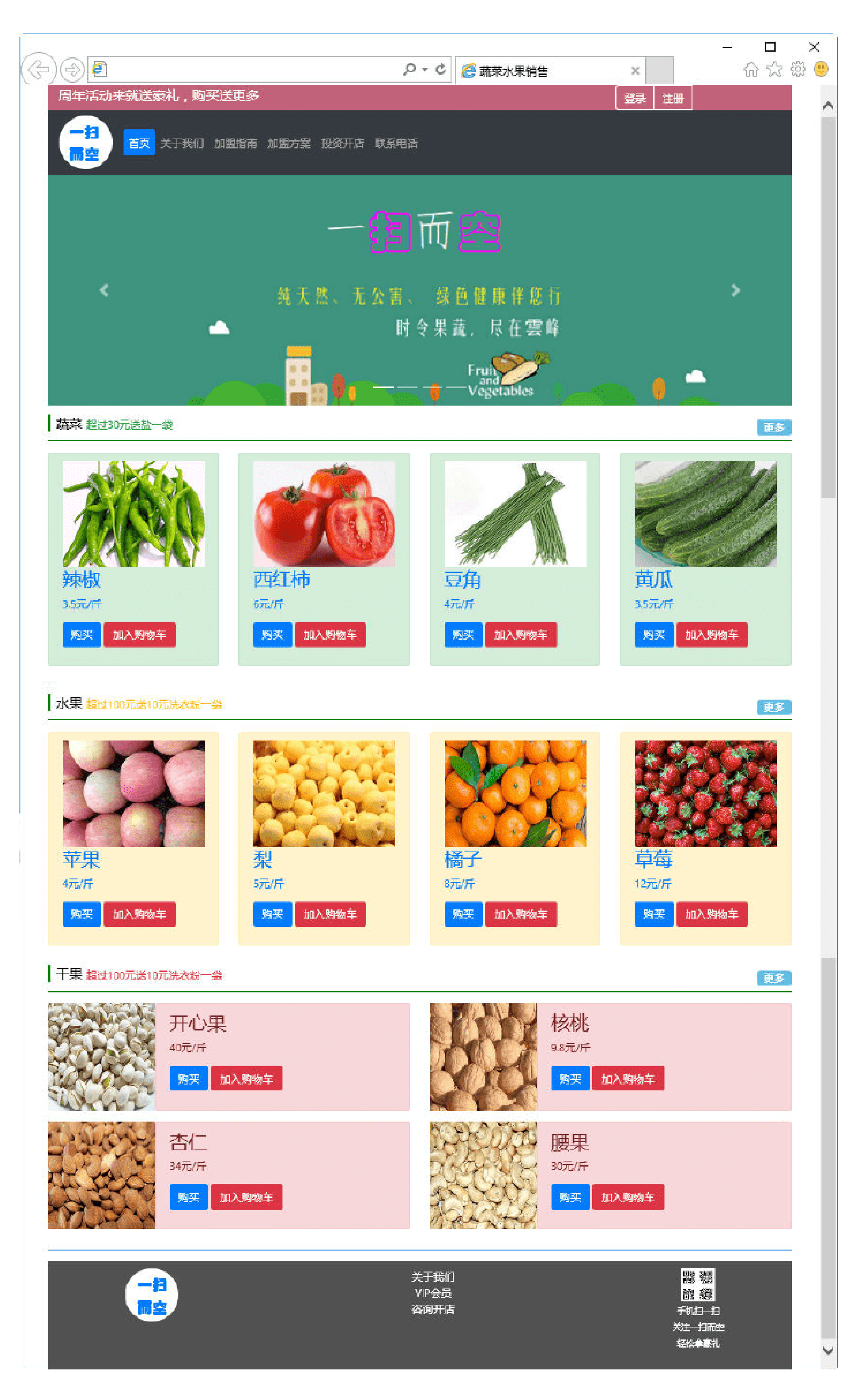


图 23-2　首页页面效果

在广告栏中，单击“登录”按钮时会弹出模态框，如图 23-3 所示；单击“注册”按钮时会弹出模态框，如图 23-4 所示。

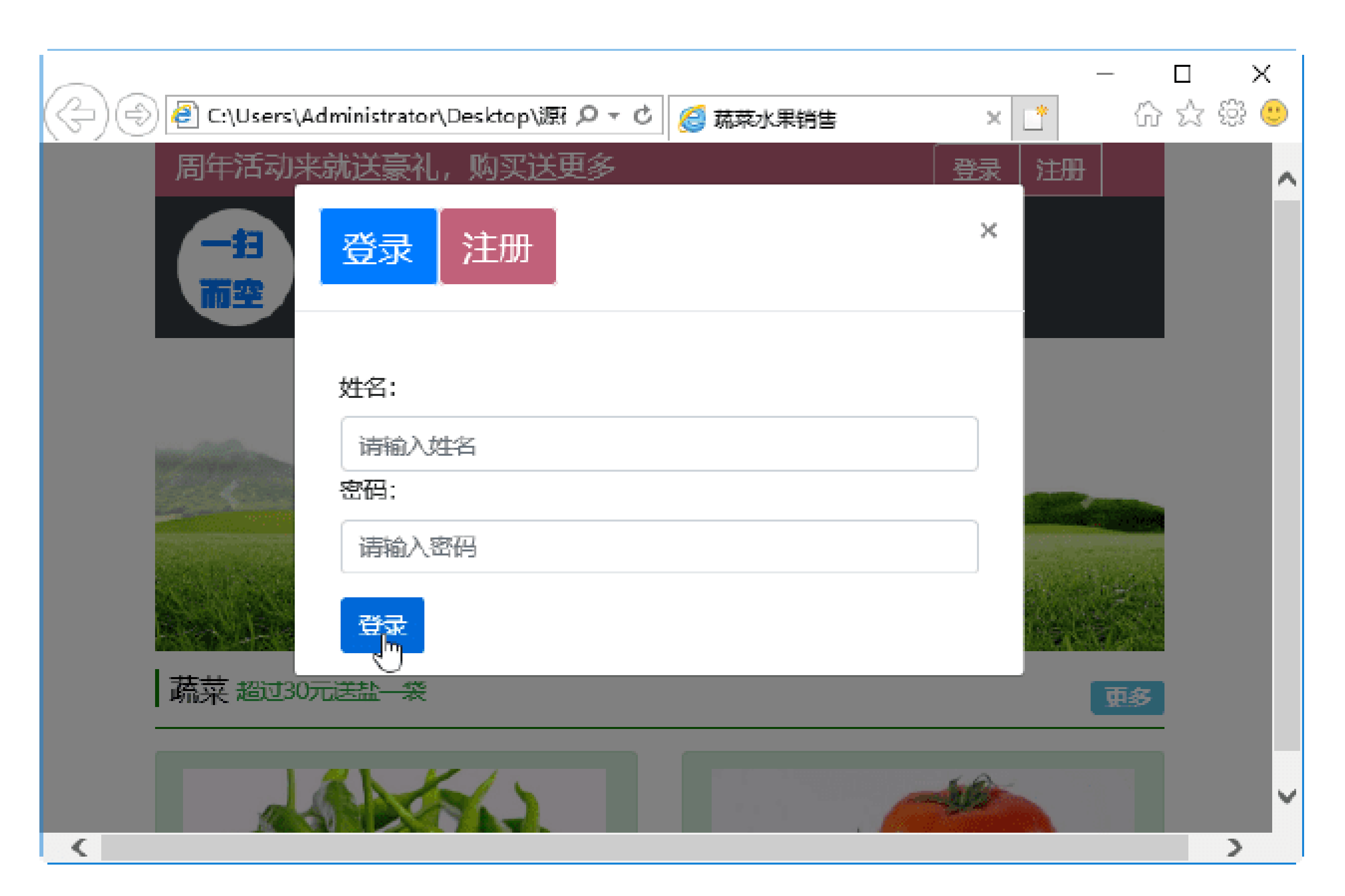


图 23-3　登录页面

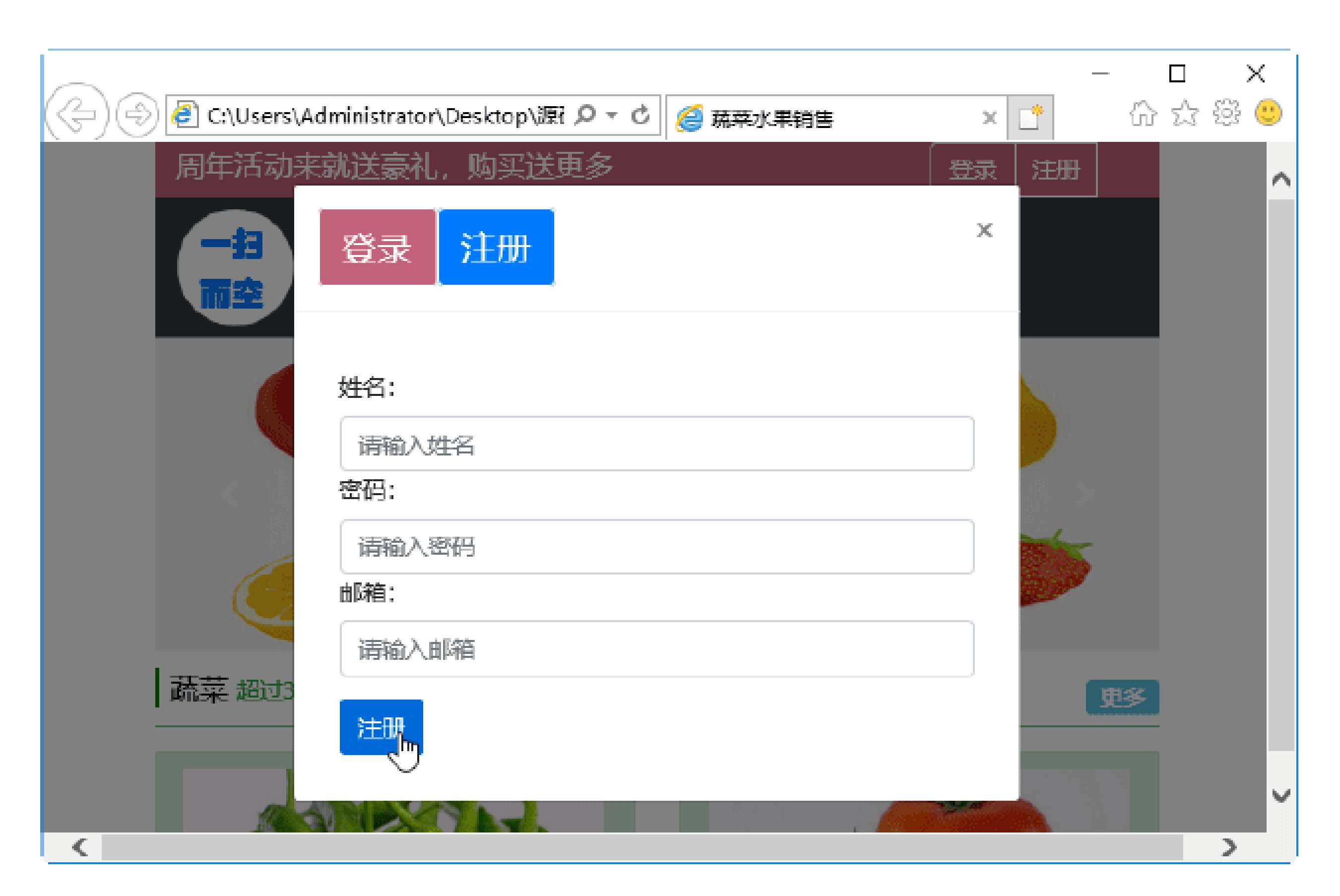


图 23-4　注册页面

在蔬菜栏中单击“购买”按钮时将跳转到购买页面，如图 23-5 所示。

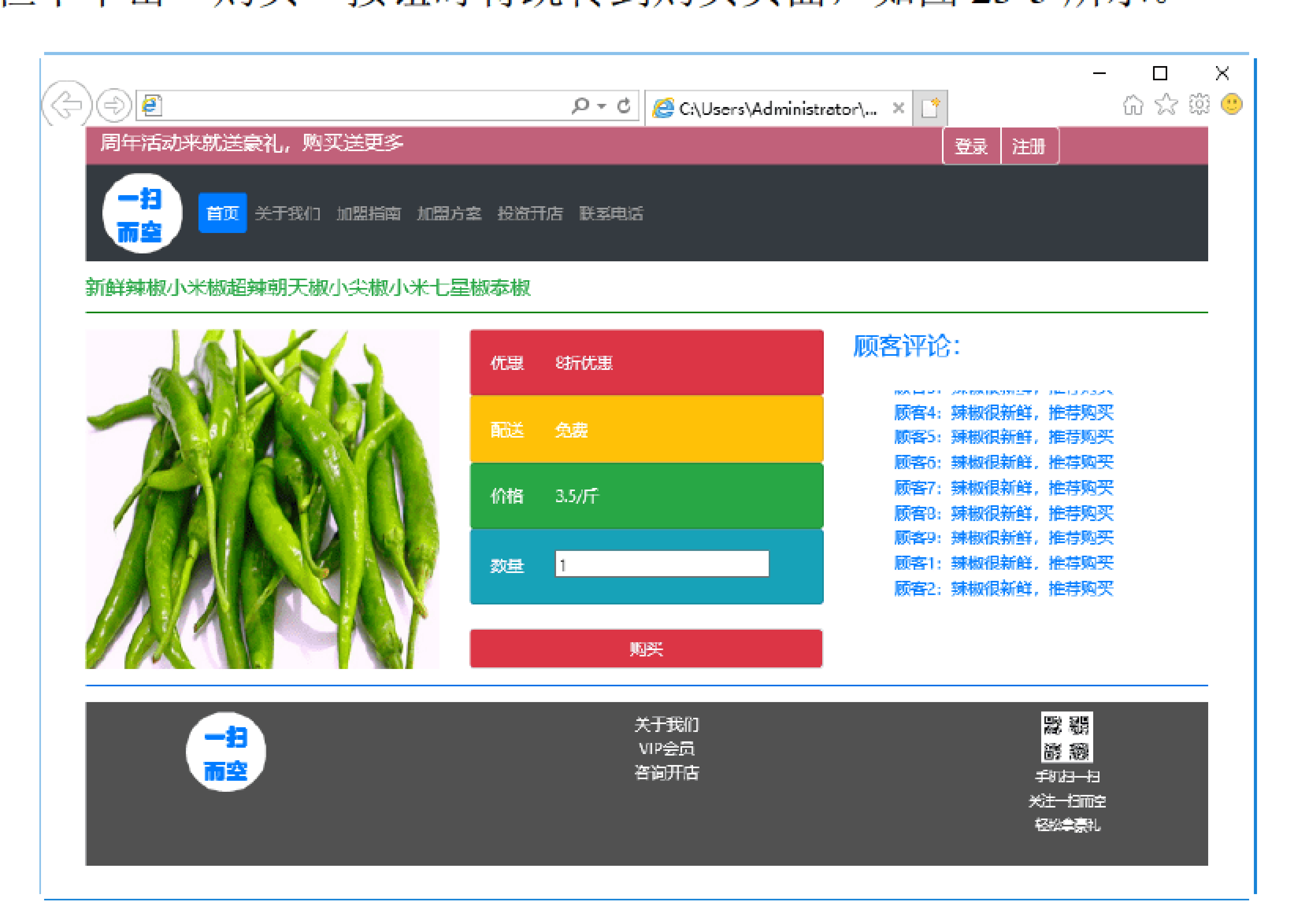


图 23-5　购买页面

选择好斤数，单击“购买”按钮，将弹出一个模态框，模态框中显示购买辣椒的总钱数，单击“确认购买”按钮，即可购买辣椒，如图 23-6 所示。

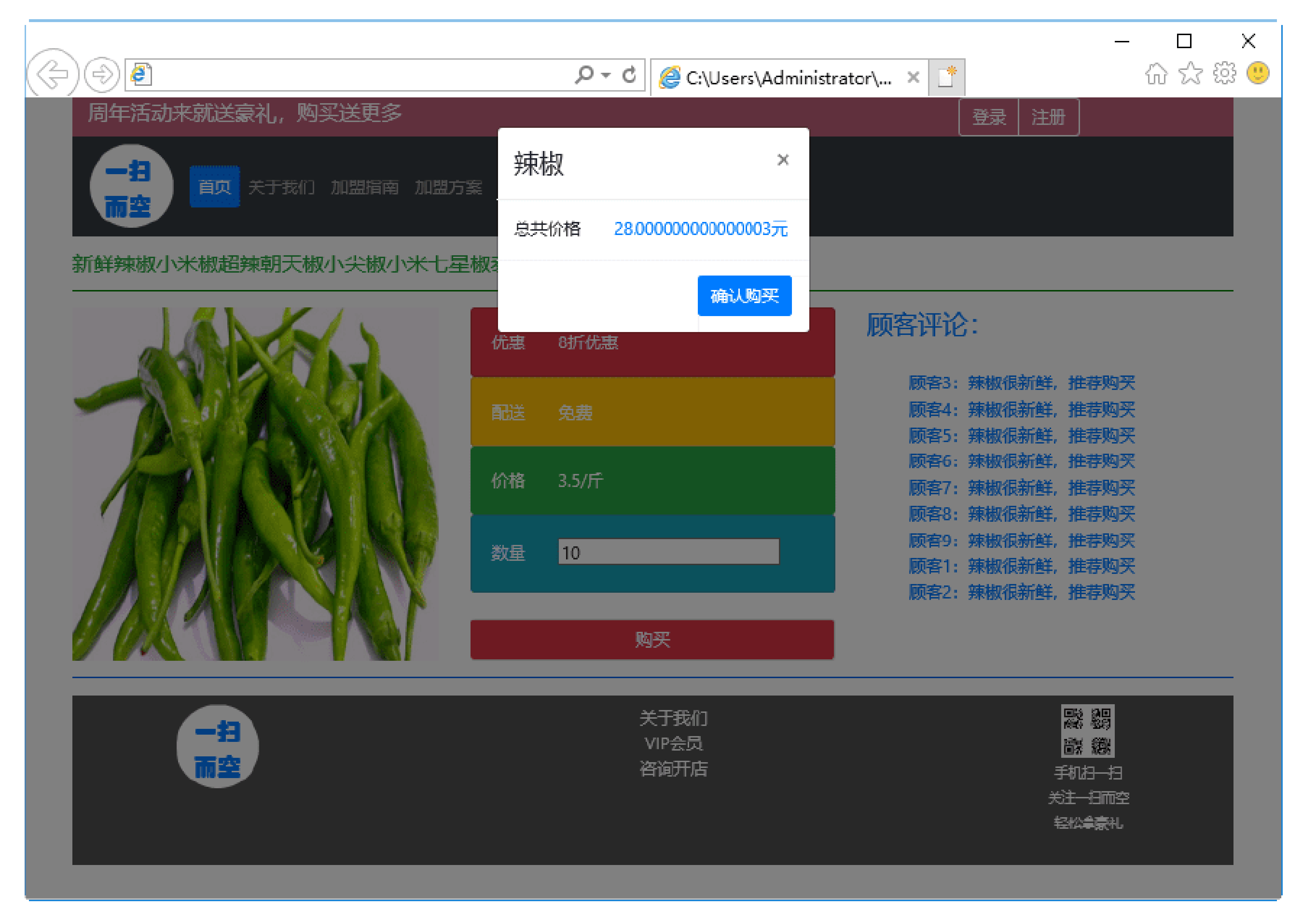


图 23-6　购买效果

在首页中，单击蔬菜栏中的“更多”按钮时，将跳转到所有蔬菜的展示页面，如图 23-7 所示。在该页面中也可以完成购买。

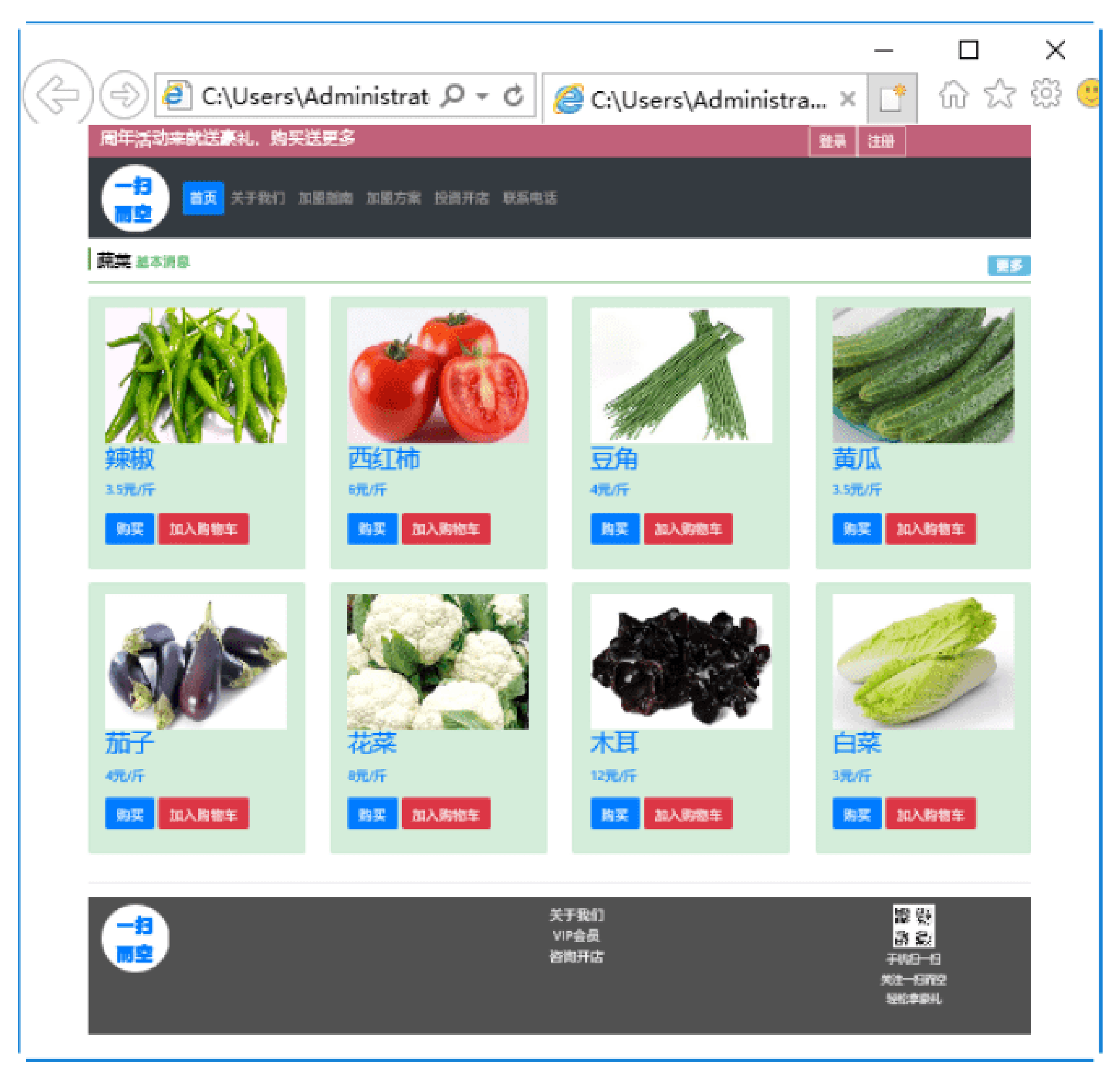


图 23-7　蔬菜展示效果

23.2　首页设计

首页的设计很重要，它会直接影响网站的受欢迎程度，下面就来具体介绍本网站的设计思路。

在介绍之前，首先需要在头部引入 Bootstrap 框架的文件和 jQuery 文件，如下面的代码所示。

```
    <head>
    <title>蔬菜水果销售</title>
        <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
        <link rel="stylesheet" href="bootstrap-4.1.3/css/bootstrap.css">
        <link rel="stylesheet" href="css/style.css">
        <script src="js/jquery.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.bundle.js"></script>
        <script src="bootstrap-4.1.3/js/bootstrap.js"></script>
    </head>
```

注意，Bootstrap 中的页面内容和栅格系统需要包裹在特定的容器中。Bootstrap 提供了两个类，分别为.container 和.container-fluid。.container 类用于固定宽度并支持响应式布局的容器；.container-fluid 类用于占据全部视口（viewport）的宽度。本项目使用含有.container 类的容器。

23.2.1　设计广告栏

广告栏采用了 Bootstrap 的网格系统，页面效果如图 23-8 所示。

周年活动来就送豪礼，购买送更多　登录　注册

图 23-8　在大于 768 像素宽度屏幕上的效果

Bootstrap 网格系统的布局是响应式的，页面中的列会根据屏幕大小自动重新排列。

Bootstrap 4 网格系统有以下 5 个类。

（1）col-：针对所有设备。

（2）col-sm-：针对平板设备（屏幕宽度等于或大于 576 像素）。

（3）col-md-：针对桌面显示器（屏幕宽度等于或大于 768 像素）。

（4）col-lg-：针对大桌面显示器（屏幕宽度等于或大于 992 像素）。

（5）col-xl-：针对超大桌面显示器（屏幕宽度等于或大于 1200 像素）。

Bootstrap 网格系统在不同设备上的情况如表 23-1 所示。

表 23-1　网格系统在不同设备上的情况

	超小设备<576 像素	平板≥576 像素	桌面显示器≥768 像素	大桌面显示器≥992 像素	超大桌面显示器≥1200 像素
容器最大宽度	None(auto)	540 像素	720 像素	960 像素	1140 像素
类前缀	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
列数量和	12				
间隙宽度	30 像素 （一个列的每边分别 15 像素）				

广告栏中有两部分信息：左边是广告信息；右边是登录注册信息。两部分根据屏幕的大小，会占据不同的空间。左边设置为“col-xs-12 col-sm-12 col-md-9 col-lg-9”，右边设置为“col-xs-12 col-sm-12 col-md-3 col-lg-3”。当屏幕宽度大于 768 像素时，左边将占据网格的 9 份，右边占据 3 份；当屏幕宽度小于 768 像素时，左边和右边都占据 12 份，在两行显示，效果如图 23-9 所示。

在右边部分，为登录注册定义了一个模态框，用来让读者进行注册或登录。在模态框中又添加了一个胶囊导航选项卡，用来切换登录和注册。



图 23-9　在小于 768 像素宽度屏幕上的效果

广告栏具体的代码如下所示。

CSS 样式代码如下：

```
.hot{margin: 0;background:#C1617A;}
.btn-group a{
    background: #C1617A;
    color: white!important;
    border: 1px solid white;
}
```

HTML 代码如下：

```
<div class="row hot">
        <div class="col-xs-12 col-sm-12 col-md-9 col-lg-9">
            <span style="color: white;font-size: 20px;font-family:微软雅
黑;">周年活动来就送豪礼，购买送更多</span>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-3 col-lg-3">
            <div class="btn-group">
                <a type="button" class="btn btn-default" data-toggle="modal"
```

```
data-target="#myModal">登录</a>
                    <a type="button" class="btn btn-default" data-toggle="modal"
data-target="#myModal">注册</a>
                        <!--开始演示模态框-->
                    <!-- 模态框（Modal） -->
                    <div class="modal fade" id="myModal" tabindex="-1"
role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
                        <!--定义模态对话框层-->
                        <div class="modal-dialog">
                            <div class="modal-content">
                                <div class="modal-header">
                                    <h4 class="modal-title" id="myModalLabel">
                                        <!--胶囊导航选项卡切换-->
                                        <ul class="nav nav-pills" role="tablist">
                                            <li class="nav-item">
                                                <a class="nav-link active"
data-toggle="pill" href="#home">登录</a>
                                            </li>
                                            <li class="nav-item">
                                                <a class="nav-link"
data-toggle="pill" href="#menu1">注册</a>
                                            </li>
                                        </ul>
                                    </h4>
                                    <button type="button" class="close"
data-dismiss="modal">&times;</button>
                                </div>
                                <div class="modal-body">
                                    <!--胶囊选项卡-->
                                    <div class="tab-content">
                                        <div id="home" class="container tab-pane
active"><br>
                                            <form role="form">
                                                <div class="form-group">
                                                    <label for="name">姓名：</label>
                                                    <input type="text"
class="form-control" id="name" placeholder="请输入姓名">
                                                    <label for="name1">密码：</label>
                                                    <input type="password"
class="form-control" id="name1" placeholder="请输入密码">
                                                </div>
                                                <a type="button" class="btn
btn-primary">登录</a>
                                            </form>
                                        </div>
                                        <div id="menu1" class="container tab-pane
fade"><br>
                                            <form role="form">
```

```
                                    <div class="form-group">
                                        <label for="name2">姓名：</label>
                                        <input type="text"
class="form-control" id="name2" placeholder="请输入姓名">
                                        <label for="name3">密码：</label>
                                        <input type="password"
class="form-control" id="name3" placeholder="请输入密码">
                                        <label for="name4">邮箱：</label>
                                        <input type="email"
class="form-control" id="name4" placeholder="请输入邮箱">
                                    </div>
                                    <a type="button" class="btn
btn-primary">注册</a>
                                </form>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

23.2.2　设计导航栏

本项目使用了折叠导航栏，效果如图 23-10 和图 23-11 所示。通常情况下，小屏幕上都会折叠导航栏，通过点击来显示导航选项。



图 23-10　大屏幕下导航效果

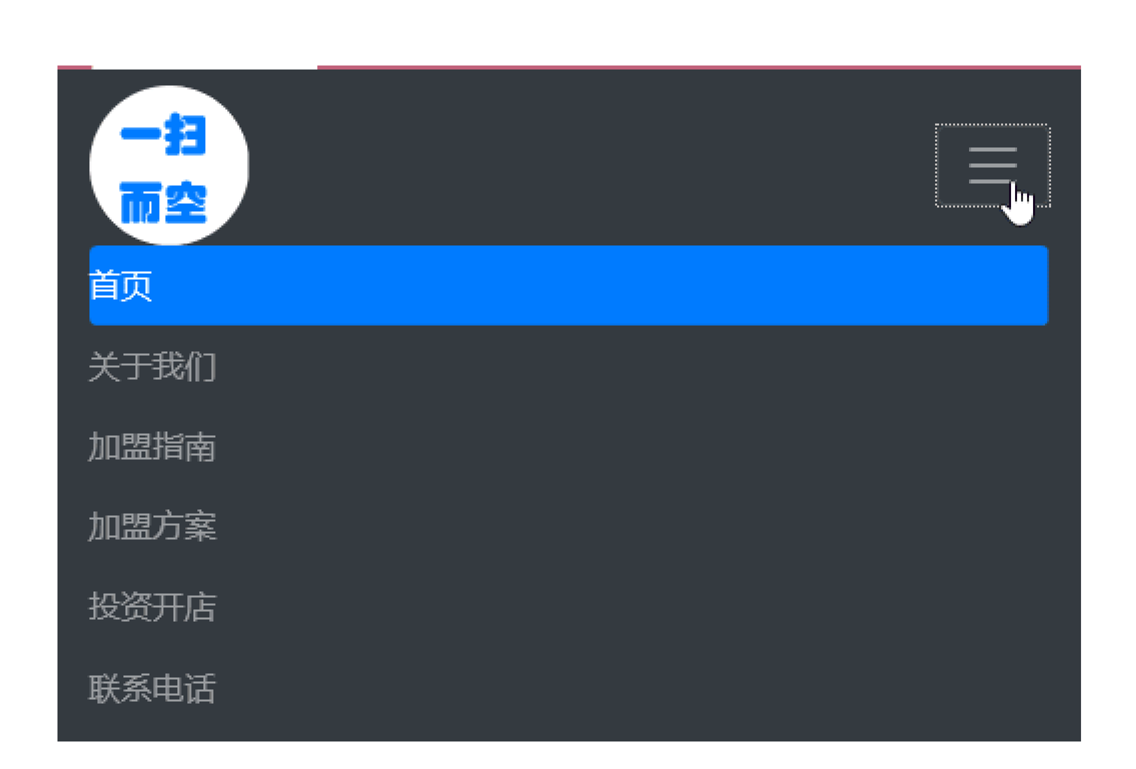


图 23-11　小屏幕下导航效果

要创建折叠导航栏，可以在按钮上添加 class="navbar-toggler"、data-toggle="collapse"与

data-target="#thetarget"类，然后在设置了 class="collapse navbar-collapse"类的 div 上包裹导航内容（链接），div 元素上的 id 匹配按钮 data-target 上指定的 id。

Logo 样式的代码如下：

```
    .big{
        width: 80px;
        height: 80px;
        font-size: 1.4em;
        border-radius:50% 50%;
        padding: 8px 15px;
        background: white;
        font-family:华文琥珀;
    }
```

HTML 的代码如下：

```
    <nav class="navbar navbar-expand-md bg-dark navbar-dark nav-css">
           <div class="big"><a href="index.html">一扫而空</a></div>
           <a class="navbar-brand" href="#">
           </a>
           <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#collapsibleNavbar">
              <span class="navbar-toggler-icon"></span>
           </button>
           <div class="collapse navbar-collapse" id="collapsibleNavbar">
              <ul class="nav navbar-nav nav-pills">
                 <li class="nav-item"><a class="nav-link active" href="#"
data-toggle="pill">首页</a></li>
                 <li class="nav-item"><a class="nav-link" href="#"
data-toggle="pill">关于我们</a></li>
                 <li class="nav-item"><a class="nav-link" href="#"
data-toggle="pill">加盟指南</a></li>
                 <li class="nav-item"><a class="nav-link" href="#"
data-toggle="pill">加盟方案</a></li>
                 <li class="nav-item"><a class="nav-link" href="#"
data-toggle="pill">投资开店</a></li>
                 <li class="nav-item"><a class="nav-link" href="#"
data-toggle="pill">联系电话</a></li>
              </ul>
           </div>
    </nav>
```

23.2.3 设计轮播

本项目中的轮播图效果如图 23-12 所示。



图 23-12　轮播效果

Bootstrap 中轮播是一种灵活的响应式插件。除此之外，内容还可以是图像、内嵌框架、视频或其他想要放置的任何类型内容。Bootstrap 中与轮播相关的类别属性如下所示：

（1）.carousel：创建一个轮播图。

（2）.carousel-indicators：为轮播图添加一个指示符，就是轮播图底下的一个个小点，轮播的过程可以显示目前是第几张图。

（3）.carousel-inner：添加要切换的图片。

（4）.carousel-item：指定每个图片的内容。

（5）.carousel-control-prev：添加左侧的按钮，点击会返回上一张。

（6）.carousel-control-next：添加右侧的按钮，点击会切换到下一张。

（7）.carousel-control-prev-icon：与.carousel-control-prev 一起使用，设置左侧的按钮。

（8）.carousel-control-next-icon：与.carousel-control-next 一起使用，设置右侧的按钮。

（9）.slide：切换图片的过渡和动画效果，如果不需要这样的效果，就可以删除这个类。

下面是本项目轮播图的具体代码。

```
<div id="demo" class="carousel slide" data-ride="carousel">
      <!-- 指示符 -->
      <ul class="carousel-indicators">
         <li data-target="#demo" data-slide-to="0" class="active"></li>
         <li data-target="#demo" data-slide-to="1"></li>
         <li data-target="#demo" data-slide-to="2"></li>
         <li data-target="#demo" data-slide-to="3"></li>
      </ul>
      <!--轮播图片-->
      <div class="carousel-inner">
         <div class="carousel-item active">
            <img src="images/a.png">
         </div>
         <div class="carousel-item">
            <img src="images/d.png">
         </div>
         <div class="carousel-item">
            <img src="images/c.png">
```

```
            </div>
            <div class="carousel-item">
                <img src="images/b.png">
            </div>
        </div>
        <!-- 左右切换按钮 -->
        <a class="carousel-control-prev" href="#demo" data-slide="prev">
            <span class="carousel-control-prev-icon"></span>
        </a>
        <a class="carousel-control-next" href="#demo" data-slide="next">
            <span class="carousel-control-next-icon"></span>
        </a>
</div>
```

23.2.4 设计蔬菜栏

蔬菜栏和水果栏设计基本一样，是使用 Bootstrap 框架的警告框（alert）来设计布局的。下面以蔬菜栏为例具体介绍一下。

根据不同大小的屏幕设计每一行的列数，总共有 4 条数据，这里设计为“col-xs-12 col-sm-12 col-md-6 col-lg-3”。当屏幕宽度大于 960 像素时，列数为 4（12/3），显示效果如图 23-13 所示；当屏幕宽度在大于 768 像素且小于 960 像素时，列数为 2（12/6），显示效果如图 23-14 所示；当屏幕宽度小于 768 像素时，列数为 1，显示效果如图 23-15 所示。

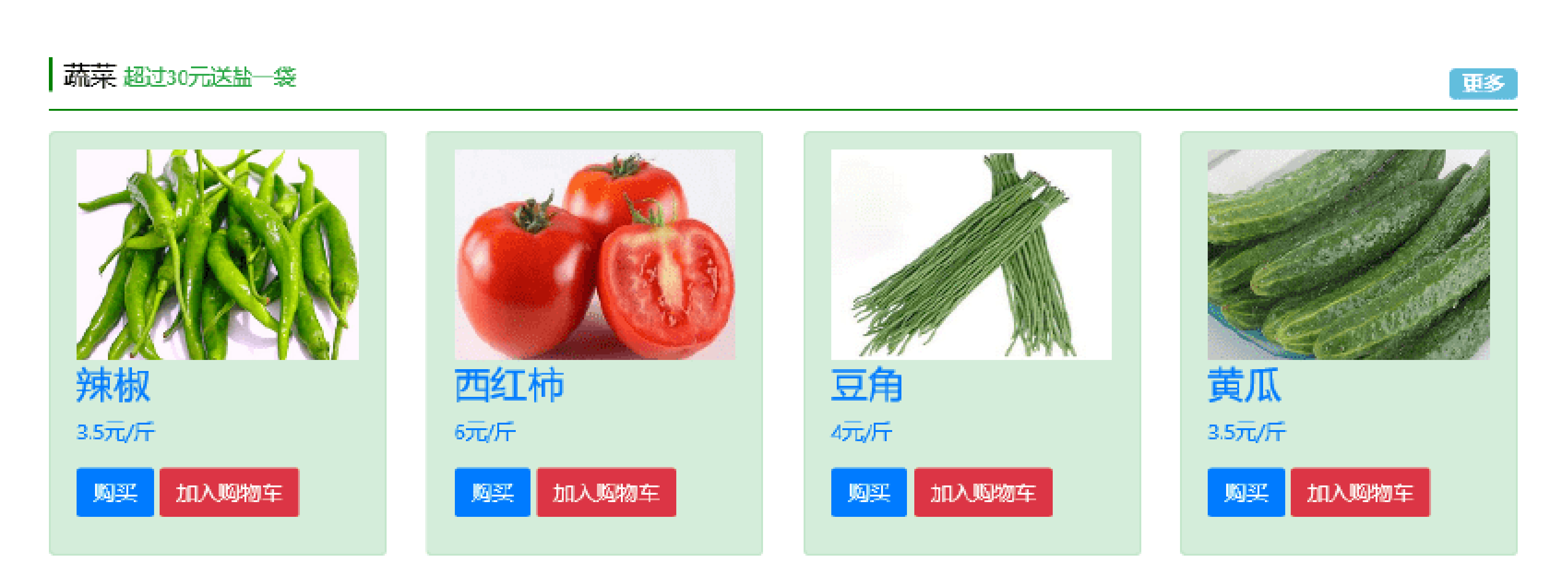


图 23-13 大于 960 像素屏幕的页面效果

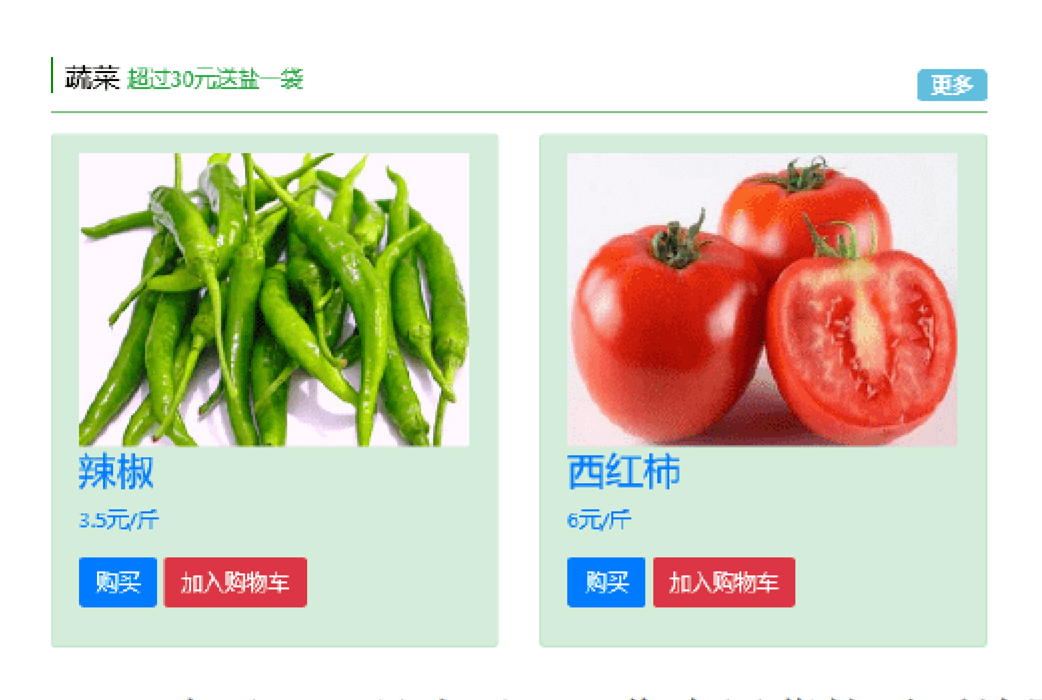


图 23-14 大于 768 且小于 960 像素屏幕的页面效果



图 23-15 小于 768 像素屏幕的页面效果

布局完成后，在每列中添加提示框（.alert 类）并添加 alert-success 类，在提示框中设计蔬菜信息，具体请参考下面代码。

CSS 样式代码如下：

```
.head-tit{
      font-size: 20px;
      line-height: 50px;
      color: black;
     border-bottom: 1px solid green;
}
.span-tit{
      border-left:3px solid green;
      padding-left: 8px;
}
.a-tit{
    background:#5bc0de;
    float: right;
    display: inline;
    padding: .2em .6em .3em;
    font-size: 80%;
    font-weight: 700;
    line-height: 1;
    color: #fff;
    border-radius: .25em;
    margin-top: 20px;
}
img{
   width: 100%;
   height: 50%;
}
```

HTML 代码如下：

```
<p class="head-tit">
      <span class="span-tit">蔬菜</span>
      <span class="text-success"><small>超过30元送盐一袋</small></span>
      <a href="buy.html" class="a-tit">更多</a>
   </p>
<div class="row">
      <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
         <a href="">
            <div class="alert alert-success">
               <img src="images/01.png" alt="辣椒">
               <div>
                  <h3 >辣椒</h3>
                  <p>3.5元/斤</p>
                  <p>
```

```
                                <a href="buy.html" class="btn btn-primary"
role="button">购买</a>
                                <a href="#" class="btn btn-danger" role="button">
加入购物车</a>
                            </p>
                        </div>
                    </div>
                </a>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
                <a href="">
                    <div class="alert alert-success">
                        <img src="images/02.png" alt="西红柿">
                        <div class="caption">
                            <h3>西红柿</h3>
                            <p>6元/斤</p>
                            <p>
                                <a href="#" class="btn btn-primary" role="button">
购买</a>
                                <a href="#" class="btn btn-danger" role="button">
加入购物车</a>
                            </p>
                        </div>
                    </div>
                </a>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
                <a href="">
                    <div class="alert alert-success">
                        <img src="images/03.png" alt="豆角">
                        <div class="caption">
                            <h3>豆角</h3>
                            <p>4元/斤</p>
                            <p>
                                <a href="#" class="btn btn-primary" role="button">
购买</a>
                                <a href="#" class="btn btn-danger" role="button">
加入购物车</a>
                            </p>
                        </div>
                    </div>
                </a>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
                <a href="">
                    <div class="alert alert-success">
                        <img src="images/04.png" alt="黄瓜">
                        <div class="caption">
```

```
                            <h3>黄瓜</h3>
                            <p>3.5元/斤</p>
                            <p>
                                <a href="#" class="btn btn-primary" role="button">
购买</a>
                                <a href="#" class="btn btn-danger" role="button">
加入购物车</a>
                            </p>
                        </div>
                    </div>
                </a>
            </div>
    </div>
```

23.2.5　设计干果栏

干果栏也是采用网格系统来设计，这里使用了网格系统的嵌套，数据有 4 条。先是外层的.row，根据不同大小的屏幕将每列设计为“col-xs-12 col-sm-12 col-md-6 col-lg-6”，然后在每一个列中嵌套一个.row。嵌套的.row 中又包含两个部分，左边是干果图片展示，根据不同大小的屏幕将该部分设计为“col-xs-12 col-sm-12 col-md-12 col-lg-4”；右边是干果的说明信息，设计为“col-xs-12 col-sm-12 col-md-12 col-lg-8”。

这样在不同大小的屏幕显示时，页面效果会自动响应，当屏幕宽度大于 960 像素时，外层的列数为 2，嵌套层左侧占 4 份，右侧占 8 份，显示效果如图 23-16 所示；当屏幕宽度大于 768 像素且小于 960 像素时，外层列数为 2，嵌套层左侧占 12 份，右侧占 12 份，将在两行显示，显示效果如图 23-17 所示；当屏幕宽度小于 768 像素时，外层列数变为 1，嵌套层左侧占 12 份，右侧占 12 份，显示效果如图 23-18 所示。

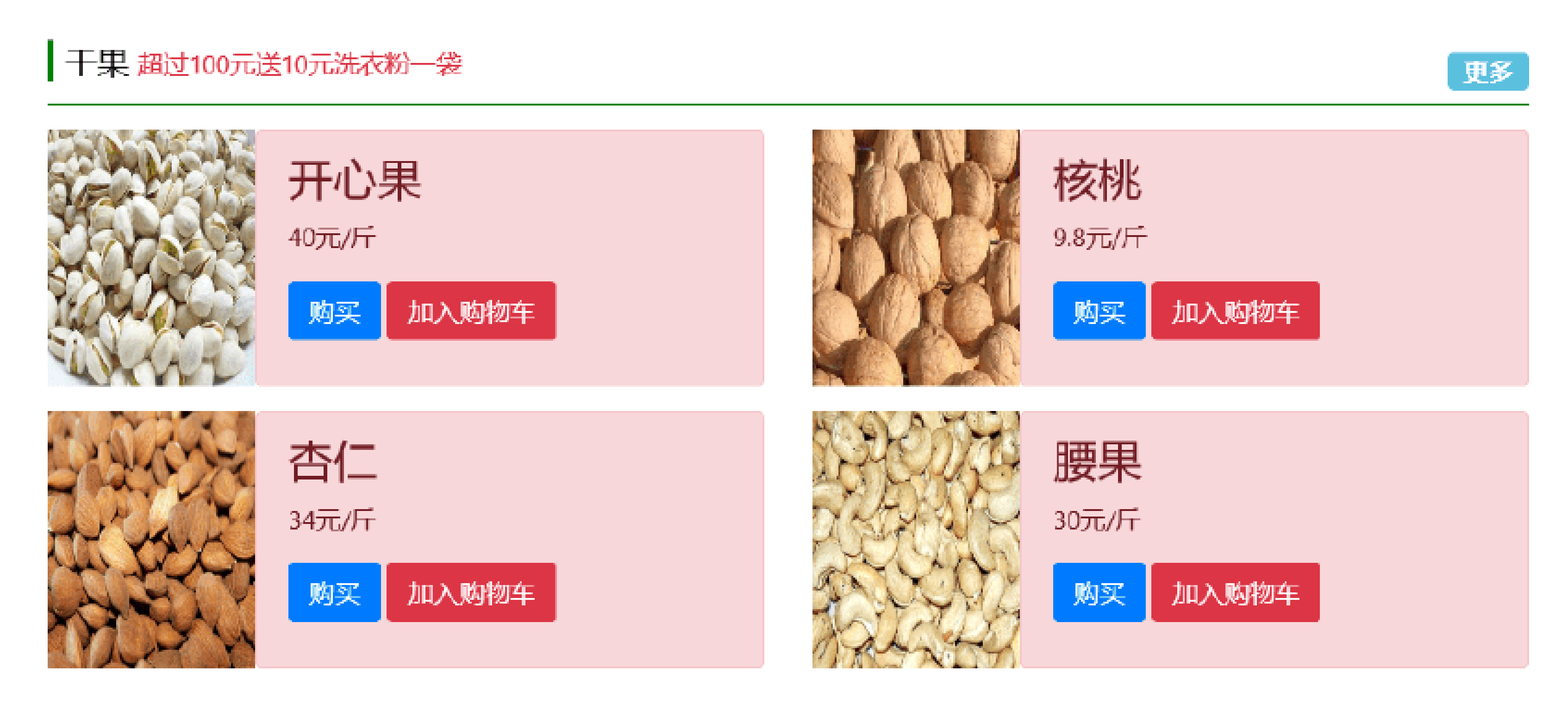


图 23-16　大于 960 像素屏幕的页面效果

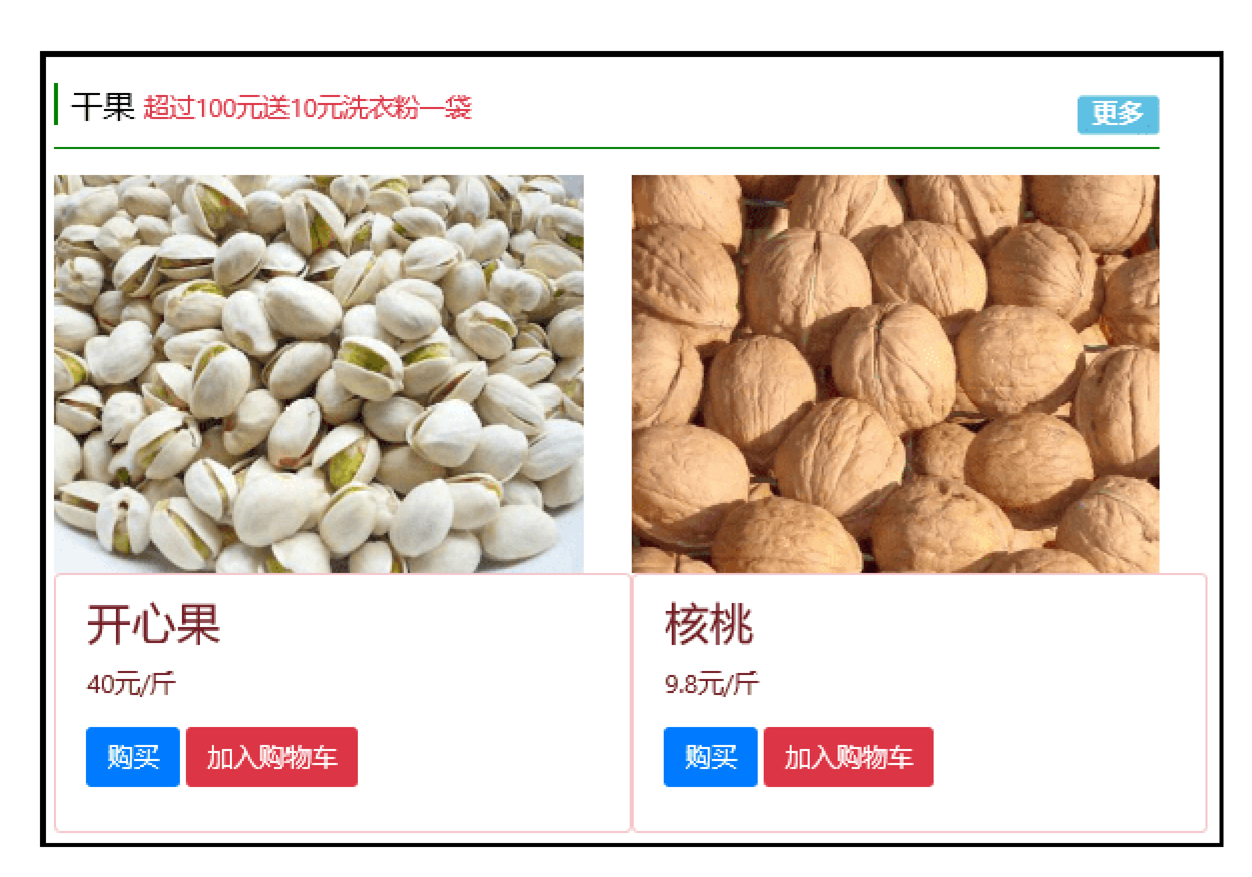


图 23-17 大于 768 且小于 960 像素屏幕的效果

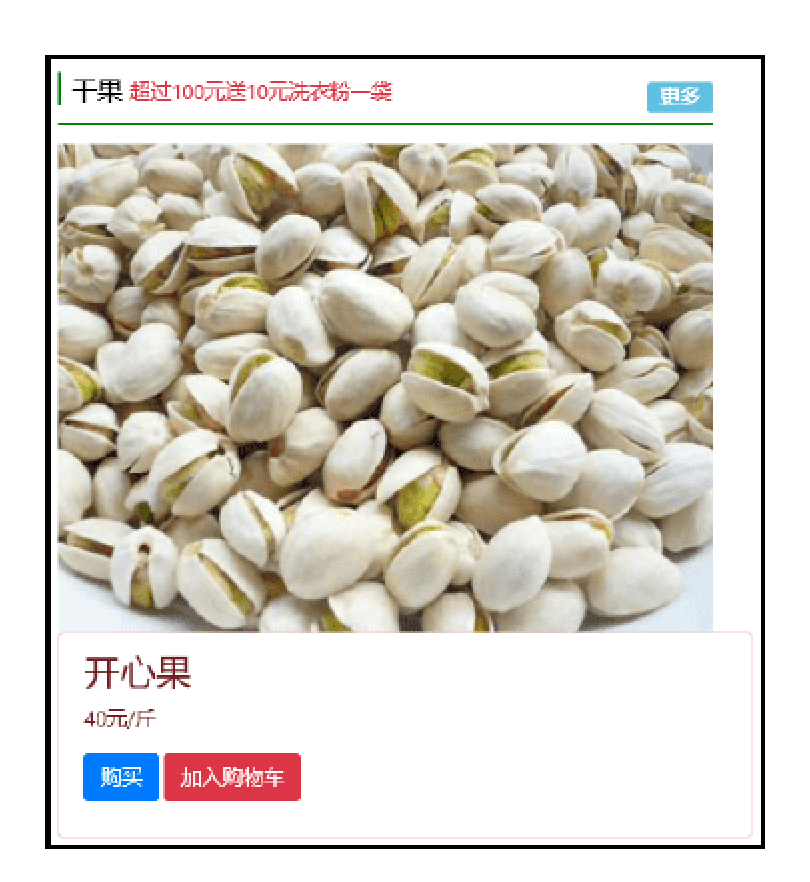


图 23-18 小于 768 像素屏幕的效果

布局完成后，在每列中添加了提示框（.alert 类）并添加 alert-dange 类。在提示框中设计干果的信息，具体请参考下面代码。

CSS 样式代码如下：

```
img{
width: 100%;
height: 50%;
}
.row-imgs img{
height:91%;
}
.row-list{
margin-left: -15px;
}
@media (max-width: 767px) {
.row-imgs img{
width: 100%;
height:100%;
}
.row-list{
margin-left: 15px;
background: white;
}
}
@media (min-width: 768px)and (max-width: 991px){
.row-imgs img{
width: 100%;
height:100%;
}
.row-list{
margin-left: 15px;
background: white;
}
```

```
    }
    .head-tit{
    font-size: 20px;
    line-height: 50px;
    color: black;
    border-bottom: 1px solid green;
    }
    .span-tit{
    border-left:3px solid green;
    padding-left: 8px;
    }
    .a-tit{
    background:#5bc0de;
    float: right;
    display: inline;
    padding: .2em .6em .3em;
    font-size: 80%;
    font-weight: 700;
    line-height: 1;
    color: #fff;
    border-radius: .25em;
    margin-top: 20px;
    }
```

HTML 代码如下：

```
    <p class="head-tit">
          <span class="span-tit">干果</span>
          <span class="text-danger"><small>超过100元送10元洗衣粉一袋
</small></span>
          <a href="" class="a-tit">更多</a>
       </p>
       <div class="row row-imgs">
          <div class="col-xs-12 col-sm-12 col-md-6 col-lg-6">
             <div class="row">
                <div class="col-xs-12 col-sm-12 col-md-12 col-lg-4">
                   <img src="images/1.png" alt="开心果">
                </div>
                <div class="col-xs-12 col-sm-12 col-md-12 col-lg-8 alert
alert-danger row-list">
                   <div class="caption">
                      <h3>开心果</h3>
                      <p>40元/斤</p>
                      <p>
                         <a href="#" class="btn btn-primary" role="button">
购买</a>
                         <a href="#" class="btn btn-danger" role="button">
加入购物车</a>
```

```
                        </p>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-6 col-lg-6">
            <div class="row ">
                <div class="col-xs-12 col-sm-12 col-md-12 col-lg-4">
                    <img src="images/2.png" alt="核桃">
                </div>
                <div class="col-xs-12 col-sm-12 col-md-12 col-lg-8 alert
alert-danger row-list">
                    <div class="caption">
                        <h3>核桃</h3>
                        <p>9.8元/斤</p>
                        <p>
                            <a href="#" class="btn btn-primary" role="button">
购买</a>
                            <a href="#" class="btn btn-danger" role="button">
加入购物车</a>
                        </p>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-6 col-lg-6">
            <div class="row">
                <div class="col-xs-12 col-sm-12 col-md-12 col-lg-4">
                    <img src="images/3.png" alt="杏仁">
                </div>
                <div class="col-xs-12 col-sm-12 col-md-12 col-lg-8 alert
alert-danger row-list">
                    <div class="caption">
                        <h3>杏仁</h3>
                        <p>34元/斤</p>
                        <p>
                            <a href="#" class="btn btn-primary" role="button">
购买</a>
                            <a href="#" class="btn btn-danger" role="button">
加入购物车</a>
                        </p>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-6 col-lg-6">
            <div class="row">
                <div class="col-xs-12 col-sm-12 col-md-12 col-lg-4">
```

```
                        <img src="images/4.png" alt="腰果">
                    </div>
                    <div class="col-xs-12 col-sm-12 col-md-12 col-lg-8 alert
alert-danger row-list">
                        <div class="caption">
                            <h3>腰果</h3>
                            <p>30元/斤</p>
                            <p>
                                <a href="#" class="btn btn-primary" role="button">
购买</a>
                                <a href="#" class="btn btn-danger" role="button">
加入购物车</a>
                            </p>
                        </div>
                    </div>
                </div>
            </div>
    </div>
```

23.2.6 设计底部栏

底部栏也是采用网格系统来布局的，总共有三部分，当屏幕宽度大于 576 像素时，左侧 logo 部分占 3 份，中间说明部分占 6 份，右侧二维码部分占 3 份，显示效果如图 23-19 所示。

当屏幕宽度小于 576 像素时，每个部分都占 12 份，在 3 行显示，显示效果如图 23-20 所示。



图 23-19　大于 576 像素屏幕的页面



图 23-20　小于 576 像素屏幕的页面效果

具体请参考下面的代码。

CSS 样式代码如下：

```
.big{
    width: 80px;
    height: 80px;
    font-size: 1.4em;
    border-radius:50% 50%;
    padding: 8px 15px;
    background: white;
    font-family:华文琥珀;
}
.footer{
     padding: 10px 0px 15px;
     background-color: #515151;
     text-align: center;
     color: #fff;
     margin: 0;
}
.imag{
     width:50px;
     height: 50px;
}
.ullist{
     list-style: none;
}
.ullist a{
      color: white;
}
```

HTML 代码如下：

```
<hr class="bg-primary">
    <div class="footer row">
        <div class="col-xs-12 col-sm-3 col-md-3 col-lg-3">
            <div class="big" style="margin:auto;">
                <a href="index.html">一扫而空</a>
            </div>
        </div>
        <div class="col-xs-12 col-sm-6 col-md-6 col-lg-6">
            <ul class="ullist" style="margin:0;padding: 0;">
                <li><a href="">关于我们</a></li>
                <li><a href="">VIP会员</a></li>
                <li><a href="">咨询开店</a></li>
            </ul>
        </div>
        <div class="col-xs-12 col-sm-3 col-md-3 col-lg-3">
            <img src="images/erweima.png" alt="二维码" class="imag">
            <p>
               <small>手机扫一扫</small><br><small>关注一扫而空</small><br>
               <small>轻松拿豪礼</small>
```

```
            </p>
        </div>
</div>
```

23.3　购买页面设计

购买页面的头部区域和底部栏与首页中的一样，这里就不再赘述了。这里主要讲一下布局、计算总钱数和顾客评价。

1. 布局购买页面

购买页面的布局仍是采用 Bootstrap 的网格系统来设计，这里主要有三部分，分别是蔬菜图片展示、购买的信息及客户的评价。根据不同大小的屏幕设计每一行的列数，总共有三部分，这里我们设计为“col-xs-12 col-sm-12 col-md-6 col-lg-4”，当屏幕宽度大于 960 像素时，列数为 3，显示效果如图 23-21 所示；当屏幕宽度大于 768 像素且小于 960 像素时，列数为 2，显示效果如图 23-22 所示；当屏幕宽度小于 768 像素时，列数为 1，显示效果如图 23-23 所示。



图 23-21　大于 960 像素屏幕的页面效果





图 23-22　大于 768 且小于 960 像素屏幕的页面效果　　图 23-23　小于 768 像素屏幕的页面效果

在购买信息列中，使用了 Bootstrap 中的卡片（card）组件，并设计不同的颜色背景及文本颜色。

2. 计算总钱数

这里使用 JS 来动态计算总钱数。当用户点击购买按钮时，总钱数将显示在设计好的模态框中。具体的 JS 代码如下：

```
$(function(){
        $("#buy").click(function(){
            var number=0;
            var b=0;
            if($("#ipt1").val()>0){
                number=$("#ipt1").val();
                b=0.8*3.5*number+"元";
                $("#ipt2").text(b)
            }
            else if($("#ipt1").val()<1){
                $("#ipt2").text("购买数量不能为负");
            }
        })
})
```

3. 顾客评价

通过 setInterval（定时器）可以将顾客的评论信息设计成自动滚动的效果。具体的 JS 代码如下：

```
$(function(){
    // 评论内容滚动
    var timer=setInterval(fn,1000);
    function fn(){
        $("#ul").animate({top:"-25px"},1000,function(){
            $("#ul").css("top",0).find("li:first").appendTo("#ul");
        })
    }
})
```

购买页面的 CSS 样式及静态页面代码如下所示。

CSS 样式代码如下：

```
.head-tit{
        font-size: 20px;
        line-height: 50px;
        color: black;
        border-bottom: 1px solid green;
}
```

```
    .size{width: 100%;height: 100%;}
    .box1{
          width:100%;
          height: 100%;
          overflow: hidden;
    }
    #ul{
         list-style: none;
         position: relative;
    }
    #ul li{height: 25px;line-height: 25px;}
```

HTML 代码如下：

```
       <p class="head-tit">
          <span class="text-success">新鲜辣椒小米椒超辣朝天椒小尖椒小米七星椒泰椒
</span>
       </p>
       <div class="row">
          <div class="col-xs-12 col-sm-12 col-md-6 col-lg-4">
             <img src="images/01.png" alt="" class="size">
          </div>
          <div class="col-xs-12 col-sm-12 col-md-6 col-lg-4">
             <div class="card bg-danger text-white">
                <div class="card-body">
                   优惠<i style="width: 2em;display: inline-block;"></i>8折优惠
                </div>
             </div>
             <div class="card bg-warning text-white">
                <div class="card-body">
                   配送<i style="width: 2em;display: inline-block;"></i>免费
                </div>
             </div>
             <div class="card bg-success text-white">
                <div class="card-body">
                   价格<i style="width: 2em;display: inline-block;"></i>3.5/斤
                </div>
             </div>
             <div class="card bg-info text-white">
                <div class="card-body">
                   数量<i style="width: 2em;display: inline-block;"></i>
                   <input type="number" value="1" id="ipt1">
                </div>
             </div><br>
             <div class="card bg-light text-white">
                <a href="#" id="buy" class="btn btn-danger" role="button"
data-toggle="modal" data-target="#myModal">购买</a>
             </div>
```

```
                <!-- 模态框 -->
                <div class="modal fade" id="myModal">
                    <div class="modal-dialog modal-sm">
                        <div class="modal-content">
                            <!-- 模态框头部 -->
                            <div class="modal-header">
                                <h4 class="modal-title">辣椒</h4>
                                <button type="button" class="close"
data-dismiss="modal">&times;</button>
                            </div>
                            <!-- 模态框主体 -->
                            <div class="modal-body">
                                总共价格<i style="width: 2em;display:
inline-block;"></i><span id="ipt2" class="text-primary"></span>
                            </div>
                            <!-- 模态框底部 -->
                            <div class="modal-footer">
                                <button type="button" class="btn btn-primary"
data-dismiss="modal">确认购买</button>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-12 col-lg-4 text-primary">
                <h4>顾客评论：</h4><br>
                <div class="box1">
                    <ul id="ul">
                        <li><p>顾客1：辣椒很新鲜，推荐购买</p></li>
                        <li><p>顾客2：辣椒很新鲜，推荐购买</p></li>
                        <li><p>顾客3：辣椒很新鲜，推荐购买</p></li>
                        <li><p>顾客4：辣椒很新鲜，推荐购买</p></li>
                        <li><p>顾客5：辣椒很新鲜，推荐购买</p></li>
                        <li><p>顾客6：辣椒很新鲜，推荐购买</p></li>
                        <li><p>顾客7：辣椒很新鲜，推荐购买</p></li>
                        <li><p>顾客8：辣椒很新鲜，推荐购买</p></li>
                        <li><p>顾客9：辣椒很新鲜，推荐购买</p></li>
                    </ul>
                </div>
            </div>
        </div>
```

23.4 蔬菜展示页面设计

蔬菜展示页面的布局与首页中的蔬菜栏布局是一样的，只是展示了所有的蔬菜。蔬菜展示页面中的广告栏、导航栏和底部栏与首页是一样的设计。具体代码如下：

```
    <p class="head-tit">
           <span class="span-tit">蔬菜</span>
           <span class="text-success"><small>超过30元送盐一袋</small></span>
           <a href="" class="a-tit">更多</a>
       </p>
       <div class="row show" id="row1">
           <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
               <a href="">
                   <div class="alert alert-success">
                       <img src="images/01.png" alt="辣椒">
                       <div class="caption">
                           <h3>辣椒</h3>
                           <p>3.5元/斤</p>
                           <p><a href="buy.html" class="btn btn-primary"
role="button">购买</a> <a href="#" class="btn btn-danger" role="button">加入购
物车</a></p>
                       </div>
                   </div>
               </a>
           </div>
           <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
               <a href="">
                   <div class="alert alert-success">
                       <img src="images/02.png" alt="西红柿">
                       <div class="caption">
                           <h3>西红柿</h3>
                           <p>6元/斤</p>
                           <p><a href="#" class="btn btn-primary" role="button">
购买</a> <a href="#" class="btn btn-danger" role="button">加入购物车</a></p>
                       </div>
                   </div>
               </a>
           </div>
           <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
               <a href="">
                   <div class="alert alert-success">
                       <img src="images/03.png" alt="豆角">
                       <div class="caption">
                           <h3>豆角</h3>
                           <p>4元/斤</p>
                           <p><a href="#" class="btn btn-primary" role="button">
购买</a> <a href="#" class="btn btn-danger" role="button">加入购物车</a></p>
                       </div>
                   </div>
               </a>
           </div>
           <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
               <a href="">
```

```
                <div class="alert alert-success">
                    <img src="images/04.png" alt="黄瓜">
                    <div class="caption">
                        <h3>黄瓜</h3>
                        <p>3.5元/斤</p>
                        <p><a href="#" class="btn btn-primary" role="button">
购买</a> <a href="#" class="btn btn-danger" role="button">加入购物车</a></p>
                    </div>
                </div>
            </a>
        </div>
    </div>

    <div class="row hidden" id="row2">
        <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
            <a href="">
                <div class="alert alert-success">
                    <img src="images/05.png" alt="辣椒">
                    <div class="caption">
                        <h3>茄子</h3>
                        <p>4元/斤</p>
                        <p><a href="#" class="btn btn-primary" role="button">
购买</a> <a href="#" class="btn btn-danger" role="button">加入购物车</a></p>
                    </div>
                </div>
            </a>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
            <a href="">
                <div class="alert alert-success">
                    <img src="images/06.png" alt="西红柿">
                    <div class="caption">
                        <h3>花菜</h3>
                        <p>8元/斤</p>
                        <p><a href="#" class="btn btn-primary" role="button">
购买</a> <a href="#" class="btn btn-danger" role="button">加入购物车</a></p>
                    </div>
                </div>
            </a>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
            <a href="">
                <div class="alert alert-success">
                    <img src="images/07.png" alt="豆角">
                    <div class="caption">
                        <h3>木耳</h3>
                        <p>12元/斤</p>
                        <p><a href="#" class="btn btn-primary" role="button">
```

```
购买</a> <a href="#" class="btn btn-danger" role="button">加入购物车</a></p>
                    </div>
                </div>
            </a>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-6 col-lg-3">
            <a href="">
                <div class="alert alert-success">
                    <img src="images/08.png" alt="黄瓜">
                    <div class="caption">
                        <h3>白菜</h3>
                        <p>3元/斤</p>
                        <p><a href="#" class="btn btn-primary" role="button">
购买</a> <a href="#" class="btn btn-danger" role="button">加入购物车</a></p>
                    </div>
                </div>
            </a>
        </div>
    </div>
```

23.5　项目打包成 APP

具体的打包过程请参考第 22 章的“22.3 项目打包成 APP”小节。

在手机上安装成功后，打开 APP，页面效果与前面项目效果展示基本一致，效果如图 23-24~图 23-26 所示。



图 23-24　主页



23-25　主页面导航栏



23-26　登录注册

第24章　项目实训4——开发游戏APP

本章将介绍一款经典的小游戏《打地鼠》，它是由 HTML5、CSS 和 JavaScript 等语言技术共同完成的。在小游戏开发中，主要使用了 HTML5 中的 canvas 元素。canvas 元素可以使浏览器直接创建并处理图像，减轻开发人员的负担；还可以使界面更加优美，提高用户体验。HTML5 开发网站和游戏已经不容忽视，不论动画细节还是运行效率都很棒，将来会有很好的前景。下面具体介绍一下小游戏《打地鼠》。

24.1　游戏概述

《打地鼠》游戏是一个趣味性的小游戏，游戏开始后，地鼠会从一个个地洞中随意出现，出现的时间有限，要在限定的时间内把它消除，每消除一个加一分，若没在限制的时间内消除地鼠，则地鼠逃生，你将失去一条命，每一次游戏共有三条命。当有三个地鼠逃生时，游戏结束。

24.1.1　游戏结构目录

本项目的目录结构如图 24-1 所示。

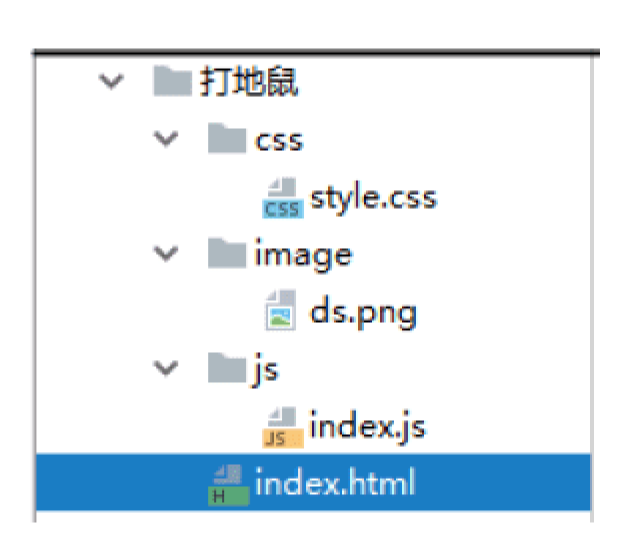


图 24-1　目录结构

具体内容如下。

（1）css 文件夹：包含项目的样式文件 style.css。

（2）image 文件夹：项目使用的图片。

（3）js 文件夹：包含项目的 JS 文件，index.js 文件包含游戏规则的具体设计内容。

（4）index.html：游戏主页面。

24.1.2 项目效果演示

本项目使用 Opera Mobile Emulator 模拟器来展示效果。打开模拟器，选择“Amazon Kindle Fire”启动，把 index.html 文件拖入模拟器，游戏直接开始，页面效果如图 24-2 所示。这时只需要用鼠标点击出现的地鼠以消除，每消除一只地鼠加一分，如图 24-3 所示。随着消除地鼠的增加，地鼠出现的间隔时间会越来越短，游戏难度也会增加，游戏得分就各凭本事了。当三次机会都用使用完后，会弹出消除的地鼠数量，如图 24-4 所示。



图 24-2 游戏主页面



24-3 开始游戏



图 24-4 游戏结束

24.2 游戏设计

《打地鼠》游戏是在 Opera Mobile 模拟器上调试开发的。实现《打地鼠》游戏，用到了 HTML5、JavaScript、CSS3、canvas 等技术。下面来看一下实现的代码。

24.2.1 index.html 文件

index.html 是项目的主页面，代码比较简洁。具体的代码如下：

```
    <!DOCTYPE html>
    <html>
    <head>
        <meta name="viewport" content="initial-scale=1, maximum-scale=1,
user-scalable=no">
```

```
    <meta charset="UTF-8">
    <title>HTML5打地鼠</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
<h2>打地鼠游戏</h2>
<h3>得分:0</h3>
<h3>生命:3</h3>
<div></div>
<canvas id="myCanvas"></canvas>
</body>
<script src="js/index.js"></script>
</html>
```

24.2.2 style.css 文件

style.css 是项目的样式文件，具体的代码如下：

```
*{
    padding: 0;
    margin: 0;
}
body{
    text-align: center;
    background-color: cornsilk;
    overflow: hidden;
}
h2{
    font-size: 40px;
    margin-top: 50px;
}
h3{
    margin-top:20px;
    color: #4f5bff;
}
img{
    position: absolute; /*绝对定位*/
    width: 33.33%;
    max-width: 300px; /*最大宽度*/
    max-height: 300px; /*最大高度*/
    transform: scale(0);
    -Webkit-transform: scale(0);
    transition: all .5s ease-out;
    -Webkit-transition: all .5s ease-out;/*地鼠出现的动画*/
}
.active{
    transform: scale(1);
    -Webkit-transform: scale(1);
```

```
}
canvas,div{
    position: absolute;
    left: 50%;
    width: 72%;
    height: auto;
    max-width: 400px;
    max-height: 400px;
    transform: translate(-50%,0%);
    -Webkit-transform: translate(-50%,0%);/*居中*/
    margin-top: 50px;
}
div{z-index: 1;}
#temp{
    position: fixed;
    top: 200%;
    left: 200%;
    transform: scale(0.1);
    -Webkit-transform: scale(0.1);
}
```

24.2.3 index.js 文件

index.js 文件是游戏规则的设计内容，具体请看下面的代码及注释。

```
var canvas = document.getElementById("myCanvas");    //获取canvas
canvas.width = 800;     //设置canvas的宽度
canvas.height = 800;   //设置canvas的高度
var cubes = 3;
var ctx = canvas.getContext("2d"); //设置canvas的绘制环境
ctx.fillStyle = "#6fcd44";         //填充canvas的颜色

var areaSize = 800/cubes;
var cubeSize = areaSize*0.96;
ctx.translate(areaSize*0.02,areaSize*0.02);
var rats = [];   //放地鼠的数组
var scores;      //得分
var life;        //生命
var interval;    //产生地鼠的间隔时间
var t,t2;
window.onload = function(){
    drawPannel(); //游戏中的方格是用canvas画的
    initGame();    //初始化游戏
};
function initGame(){
    scores = 0;//得分
    life = 3;// 3次机会
    interval = 100;//地鼠出现的间隔时间
```

```
        document.getElementsByTagName("h3")[0].innerHTML="得分:"+scores;
        document.getElementsByTagName("h3")[1].innerHTML="生命:"+life;
        t = setInterval(function(){
            generateRats();    //产生地鼠的方法
            maintanceRats();   //维护地鼠的方法
        },interval);
    }
    function drawPannel(){//画出方格，每个方格放一个地鼠并且隐藏
        for(var i=0;i<cubes;i++){
            for(var j=0;j<cubes;j++){
                ctx.fillRect(i*areaSize,j*areaSize,cubeSize,cubeSize);//画一个
方格
                var img = new Image();
                img.src = "image/ds.png";
                img.style.left = i*33.33 + "%";
                img.style.top = j*0.3333*canvas.clientHeight + "px";
                // console.log(canvas.clientHeight);
                img.addEventListener("mousedown",clicked);
                img.addEventListener('touchstart', touched);//两种事件是为了适配
不同的移动设备
                document.getElementsByTagName("div")[0].appendChild(img);//每
个方格放地鼠
                rats.push(img);//地鼠放入队列中，用于后面维护
            }
        }
    }
    function touched(){//触摸中了
        chosen(this);
    }
    function clicked(){//点击中了
        chosen(this);
    }
    function chosen(rat){
        if(rat.className == "active"){//如果地鼠显示出来了
            rat.classList.remove("active");//隐藏
            scores ++;//加分
            document.getElementsByTagName("h3")[0].innerHTML = "得
分:"+scores;//更新显示分数
            interval -= interval*0.03>2?interval*0.03:interval*0.015;//增加游戏
难度
        }
    }
    function generateRats(){//产生地鼠的方法

if(parseInt(Math.random()*100)%parseInt(((interval/12)>2?(interval/12):2))=
=0){//产生的几率越来越大
            var ID = Math.ceil(Math.random()*8);
            if(rats[ID].className == ""){//如果没有出现
```

```
                t2 = setTimeout(function(){//则调用定时器方法，让它出现
                    rats[ID].classList.add("active");
                    rats[ID].id= interval/4;//用id表示地鼠自动消失的时间，与游戏难度
相关
                },500);
            }
        }
    }
    function maintanceRats(){//维护地鼠的方法
        var activeRats = document.getElementsByClassName("active");//获取所有
出现的地鼠
        for(var i=0;i<activeRats.length;i++){//用id表示剩余时间
            activeRats[i].id--;
            if(activeRats[i].id<0){//如果到时间了
                activeRats[i].classList.remove("active");//则当前地鼠隐藏
                life --;//掉血
                interval *= 1.08;//回退一点游戏难度
                document.getElementsByTagName("h3")[1].innerHTML = "生
命:"+life;//更新血量显示
                if(life == 0){
                    lose();     //如果生命为零，则结束游戏
                }
            }
        }
    }
    function lose(){//如果输了
        clearInterval(t);//则停止计时器，等待游戏重新开始
        clearTimeout(t2);
        setTimeout(function(){//延时一点
            alert("您输了，共打了"+scores+"只地鼠。");
            for(var i=0;i<rats.length;i++){
                rats[i].classList.remove("active");//全部地鼠隐藏
            }
            setTimeout(function(){
                initGame();//重新开始游戏
            },500);//延时，等待地鼠隐藏的动画效果结束
        },10);
    }
```

24.3 项目打包成 APP

具体的打包过程请参考第 22 章的“22.3 项目打包成 APP”小节，步骤基本一致。

在手机上安装成功后，打开 APP，页面效果与前面项目效果展示基本一致，如图 24-5~图 24-7 所示。

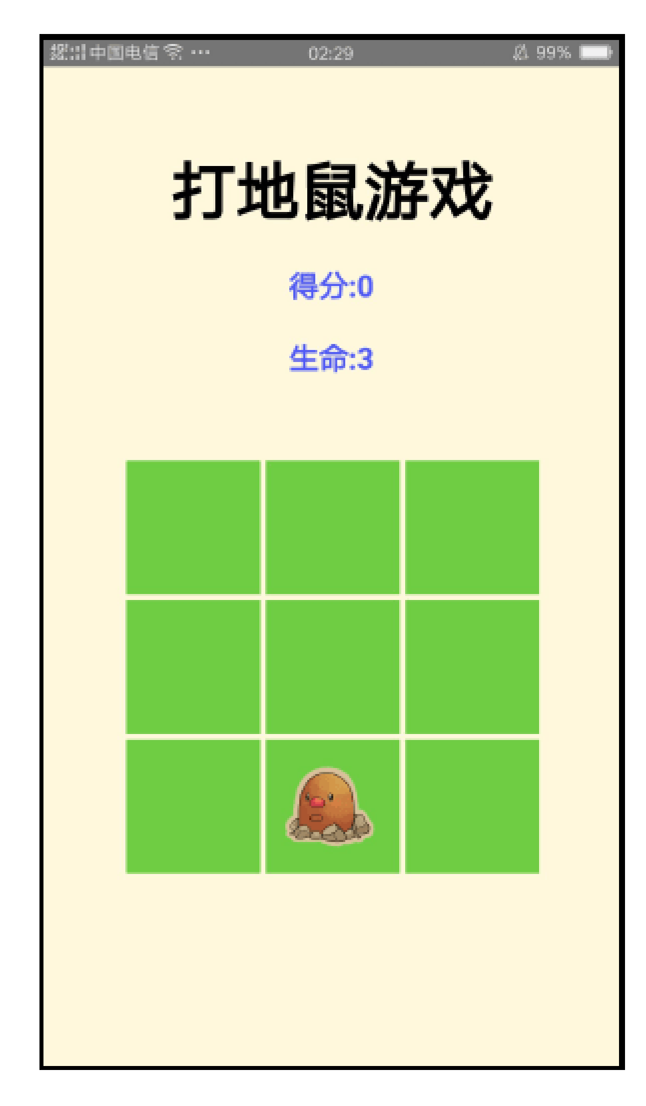



图 24-5　游戏主页面



图 24-6　开始游戏



图 24-7　结束游戏